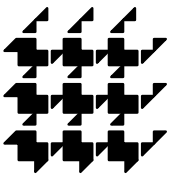
DIMACS
Center for Discrete Mathematics &
Theoretical Computer Science



DIMACS EDUCATIONAL MODULE SERIES

MODULE 04-1
Probability and Chip Firing Games

Date prepared: September 1, 2004

Lynne L. Doty
Marist College
Poughkeepsie, NY 12601
email: Lynne.Doty@Marist.edu

K. Peter Krog
Marist College
Poughkeepsie, NY 12601
email: Peter.Krog@Marist.edu

Tracey Baldwin McGrail
Marist College
Poughkeepsie, NY 12601
email: Tracey.McGrail@Marist.edu

DIMACS Center, CoRE Bldg., Rutgers University, 96 Frelinghuysen Road, Piscataway, NJ 08854-8018
TEL: 732-445-5928 • FAX: 732-445-5932 • EMAIL: center@dimacs.rutgers.edu
Web: http://dimacs.rutgers.edu/

Founded as a National Science Foundation Science and Technology Center and a Joint Project of Rutgers University, Princeton University, AT&T Labs - Research, Bell Labs, NEC Laboratories America and Telcordia Technologies with affiliated members Avaya Labs, HP Labs, IBM Research, Microsoft Research.

Module Description Information

- **Title:**

  Probability and Chip Firing Games

- **Authors:**

  Lynne L. Doty, Marist College, Poughkeepsie, NY 12601
  K. Peter Krog, Marist College, Poughkeepsie, NY 12601
  Tracey Baldwin McGrail, Marist College, Poughkeepsie, NY 12601

- **Abstract:**

  This module investigates the use of chip firing games on finite directed graphs as a "probabilistic abacus" that can be used to solve probability questions that might otherwise appear to require the use of complicated infinite geometric series. The use of chip firing games in this context leads naturally to the introduction of Markov analysis.

- **Informal Description:**

  Probability questions arising from experiments or games of chance can often be difficult to solve due to the complicated nature of the collection of possible outcomes. In certain circumstances, there might be a way to organize what could be an infinite collection of outcomes into a finite form that would offer alternate means of analysis. This module addresses these issues and offers several different approaches to solving probability questions.

- **Target Audience:**

  The intended audience consists of college students enrolled in a course on discrete mathematics or probability at the sophomore or junior level.

- **Prerequisites:**

  Students should be familiar with infinite geometric series and basic discrete probability, including expected value. Section 5 of the module requires that students also be familiar with basic matrix operations.

- **Mathematical Field:**

  Probability, Chip Firing Games, Markov analysis.

- **Applications Areas:**

  Most of the applications in this module deal with games, including both games of chance and children's board games. There is also an application to genetics in Section 5.

- **Mathematics Subject Classification:**

  Primary Classification: 91A43
  Secondary Classifications: 91A60, 60J05

- **Contact Information:**

  K. Peter Krog
  Department of Mathematics
  Marist College
  3399 North Road
  Poughkeepsie, NY 12601
  email: Peter.Krog@Marist.edu

- **Other DIMACS modules related to this module:**
  None at this time.

Contents

Abstract . . . 2

Notes to the Instructor . . . 3

Introduction . . . 4
Let's Play Lotto . . . 4

1 Which Game Should We Play? . . . 4
1.1 An Empirical Answer . . . 4
Activity . . . 5
Exercises . . . 5
1.2 Summary of Empirical Results for Game One . . . 6
1.3 An Exact Answer for Game One . . . 7
Exercises . . . 8
1.4 There's More to Life Than Lotto . . . 8
Exercises . . . 10
1.5 Calculating Probabilities Directly . . . 10
Exercises . . . 12

2 Backtracking . . . 13
2.1 Computing Probabilities by Backtracking . . . 13
Exercises . . . 15
2.2 Using Backtracking Results to Compute Expected Values . . . 16
Activity . . . 16
Exercises . . . 18

3 Calculating Probabilities Using the Addition and Multiplication Rules (Optional) 18
Exercises . . . 21

4 Chip Firing Games . . . 21
4.1 What is a Chip Firing Game? . . . 21
4.2 Tossing Coins and Firing Chips . . . 22
4.3 Chip Firing and Probability . . . 24
Exercises . . . 25
4.4 From Chips to Crumbs to Sand to Dust . . . 25

5 Coin Tossing and Chip Firing as Markov Processes . . . 26
5.1 From Chip Firing to Transition Probabilities . . . 26
5.2 Analysis of Coin Tossing Games as Markov Processes . . . 28
Activity . . . 33
Exercises . . . 34
5.3 Using Markov Analysis in More Elaborate Games . . . 35
Activity . . . 35
5.4 A Glimpse into More Standard Applications of Markov Analysis as Chip Firing Games 36
Activity . . . 37

A Solutions to Selected Exercises . . . 38

ABSTRACT

This module investigates the use of chip firing games on finite directed graphs as a "probabilistic abacus" that can be used to solve probability questions that might otherwise appear to require the use of complicated infinite geometric series. The use of chip firing games in this context leads naturally to the introduction of Markov analysis.

Notes to the Instructor

We have designed this module to run over the course of one to two weeks, depending on the level of your students and the material you choose to cover.

In Section 1, we provide motivation for the representation of games of chance using finite directed graphs. We run through several examples, first estimating the probability of winning each game and then calculating this probability using brute force and convergent series. For example, we describe the game in which the players first choose a sequence of heads or tails of length 3, then toss a coin, and record the results. Using infinite series, we are able to compute the probability that sequence HHT appears before the sequence THH.

In Section 2, we describe one way of using the finite directed graph representation of this game to compute the probability of winning the coin tossing game described in the first section. We also compute the expected number of tosses for the game. Section 3 describes an alternate approach to finding these probabilities. You can choose to replace the previous method with this one, or include both methods. Sections 2 and 3 require some basic knowledge of probability theory.

In Section 4, we use the directed graphs from the previous three sections to introduce chip firing games, a discrete variant of the games we played before. For example, we show that if we send "enough" chips through the directed graph for HHT vs. THH, the proportion of chips that end up in the terminal node HHT in the first game is exactly the probability that the sequence HHT beats the sequence THH.

In Section 5, we introduce a more sophisticated and rigorous approach to finding the probabilities described in the previous sections. We construct a transition matrix for each game and do a Markov analysis to justify the previous probability computations.

If you are intending this module for junior and senior level mathematics majors, you should be able to cover Sections 1 through 4 fairly rapidly and spend more time with the Markov analysis of Section 5. On the other hand, if you are working with freshman or sophomore level mathematics majors, or a more general audience, you may decide to omit Section 5 and proceed through the first four sections at a more leisurely pace. Sections 1 through 4 are designed to be self-contained, and to provide at the very least an intuitive understanding of why the computations we describe actually give you the desired probabilities.

Much of the content of this module was inspired by two papers written by Arthur Engel on what he called the "probabilistic abacus" (see [7, 8]). Students and instructors interested in an in-depth discussion of chip firing games may wish to refer to [3, 4, 5, 10]. Those interested in connections between chip firing games and group theory should see [2]. Algorithmic and run time issues involving chip firing games are dealt with in [15, 16]. More detailed information on Markov analysis can be found in [13, 14].

Introduction

Let's play Lotto

There's a new type of instant lotto game that is played on a video screen. Each game is played on a display like the graph shown in Figure 1. You pay $1 to play a game in which you can win $5. The game starts with your counter on the $1 node. While your counter is on any node other than the winning $5 one or the losing $0 node, the computer controlling the video screen randomly lights either the red or the green lamp in the traffic light in the lower right corner with equal probability. If the light flashes green, your counter moves to the adjacent node with the higher number. If the light flashes red, your counter moves to the adjacent node with the lower number. The game continues until your counter is on either the $0 node—you lose—or on the $5 node—you win $5. For example, if the traffic light flashed green three times in a row, you would win $5. You could also win if the light flashed two greens, a red, and a green in that order. There's one more part to the new lotto. When you pay your $1 to play new lotto, you are shown two graphs and you choose which game graph you want your counter to move on. For our example, you could play either one of the games shown in Figures 1 or 2. Each game costs $1 to play and returns $5 if your counter makes it all the way to the $5 node. So the question is: Which game gives you the better chance to win?

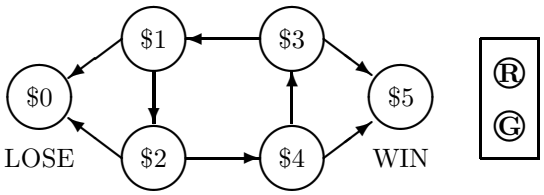


Figure 1: New Lotto Game 1

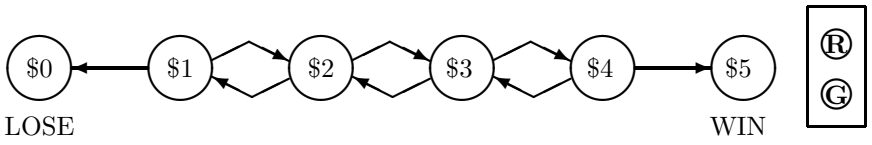


Figure 2: New Lotto Game 2

1 Which Game Should We Play?

1.1 An Empirical Answer

As a first try at determining the answer to this quandary, we could simulate the playing of many games on each diagram. As an alternative to flipping coins to simulate the traffic light, we can feed a stream of counters through the diagram according to the traffic light rules and see how many end at $5. To simulate the equally likely probabilities of moving to a higher number or moving to a lower number on the diagram, we'll use a technique that has been called the "probability abacus." If there's just one counter on a node that has arcs leading to other nodes in the graph, then the

GAME ONE

Total Chips	# Chips on Node						Prob(5)	Total Chips	# Chips on Node						Prob(5)
	0	1	2	3	4	5			0	1	2	3	4	5	
2	1		1				0	19							
4	3				1		0	21							
6	4		1		1		0	23							
8	6			1		1	.143	25							
10	7		1	1		1	.125	27							
12	9			1	1	1	.1	29							
14	10		1	1	1	1	.091	31							
16	12	1				3	.2	32							
17	13		1			3	.188	34							

Table 1: Trials of Lotto Game One

counter sits there. We don't halve it and move the pieces. Instead we wait until there are two counters on the node, then we send one to each of the adjacent nodes in the graph. The easiest way to understand this is to feed a stream of counters—one or two at a time—into one of the games from the starting point at \$1. Look at Game One (Figure 1) first. Put two counters on the \$1 node. (There's no point in putting one counter on this node because nothing happens until you have at least two on the node.) One counter moves to \$0; the other moves to \$2. Feed two more counters to node \$1; move one to \$0 and one to \$2. Now there's two counters on \$2, so to complete this phase move one to \$0 and the other to \$4. The first two lines of Table 1 show the final status of the graph after each the completion of counter movement for each of these two turns: Three counters are on \$0 and one counter is on \$4. See if you can complete the movements for feeding in a total of eight counters and check your results with the table. Notice that at the end of counter movement for eight chips one chip has finally made it to \$5 while six chips are at \$0. Then, we can use this ratio, $\frac{1}{7}$, as a first (very tentative) approximation to the probability of winning \$5.

Activity 1.1

> Divide yourselves into groups of two or three students each. Continue feeding counters into Game One to verify the entries that are given in Table 1 and to complete Table 1. Then repeat this exercise for Game Two and Table 2.

Exercises

Use your results from Activity 1.1 to answer these questions.

1. Are any of the ratios you calculated in the last column of each table the probability of winning \$5 for either game? Explain why or why not.

2. Can you estimate the probability of winning \$5 for each game using trends that appear in the tables? Write a brief justification for each estimation.

3. Given the information in the tables, which game are you more likely to win if you play it once? How much confidence do you have in your answer?

GAME TWO

Total Chips	# Chips on Node						Prob(5)	Total Chips	# Chips on Node						Prob(5)
	0	1	2	3	4	5			0	1	2	3	4	5	
2	1		1				0	16	11	1	1	1		2	.154
4	2	1		1	0		0	17							
5	3		1	1	0		0	18							
7	4	1	1		1		0	19							
8	5	1		1	1		0	21							
9	6		1	1	1		0	22							
11	7	1	1	1		1	.125	23							
12	8	1	1		1	1	.111	24							
13	9	1		1	1	1	.1	26							
14	10		1	1	1	1	.091								

Table 2: Trials of Lotto Game Two

1.2 Summary of Empirical Results for Game One

After having completed Activity 1.1 you might have noticed a few things about the patterns in the tables and the relative frequencies that you computed. Let us focus on Game One. First, you might suspect that the probability of winning \$5 in Game One is approximately 0.2. How good is this estimate? A standard technique to increase confidence in empirical probability estimates is to increase the number of trials. That's just too tedious for these games; although it might be fun to write a small program that would feed counters into the games and calculate the relative frequencies for you. A more mathematically interesting and eventually more effective way to determine the probabilities is suggested by careful analysis of some of the patterns of counters in the graph diagram. For example, you might have noticed that the relative frequency of 0.2 occurred twice in your experiments: in the rows for 16 and 31 chips. (See Table 3.)

In each case there is only one counter remaining "in play", that is, on one of the nodes with arcs leading away from it. So most of the counters you fed into the game made it all the way through to winning or losing positions, and the relative frequency of winning for these counters gives the same estimate for winning, 0.2.

Another repeating pattern of counters occurs in the row with 17 counters played and the row with 32 counters played. (See Table 4.)

While the ratios for the two rows are different (0.188 vs. 0.194, respectively), the 15 counters that moved through the game between these two rows have divided themselves between \$0 and \$5 with $25 - 13 = 12$ going to \$0 and $6 - 3 = 3$ going to \$5. Again the relative frequency of winning \$5 is $\frac{3}{15} = 0.2$. Is this phenomenon just a lucky chance for this game, or is there the germ of a

GAME ONE

Total Chips	# Chips on Node						Prob(5)
	0	1	2	3	4	5	
16	12	1				3	.2
31	24	1				6	.2

Table 3: Selected Trials of Lotto Game One

GAME ONE

Total Chips	# Chips on Node						Prob(5)
	0	1	2	3	4	5	
17	13		1			3	.188
32	25		1			6	.194

Table 4: Selected Trials of Lotto Game One

general idea here that connects the recurring patterns of counters with a correct determination of probability?

In the next few sections we'll look at three related ideas:

1. A complete description of an analytic method to calculate probability for games that are played on finite graphs,
2. An analysis of recurrent counter patterns for such games and a determination of the accuracy of probability estimates that use these patterns, and
3. A proof (using ideas from elementary absorbing Markov chains) that the method referred to in item (2) above is an exact method of determining probabilities of certain infinite probability processes that can be modeled on finite graphs.

1.3 An Exact Answer for Game One

As discussed in the previous section, we have some experimental evidence that the probability of winning \$5 in Game One is approximately 0.2. The evidence is not, however, very convincing. So in this section we'll try to determine exact answers to probability question by using mathematical analysis of the structure of the game graphs. In order to win Game One the counter must follow some path (possibly retracing some of it's earlier moves) from \$1 to \$5. Each path corresponds to a unique sequence of red(R) and green(G) flashed. For example, if the lights flash in this sequence GGGRRGGRG, the corresponding path the counter follows is 1-2-4-3-1-2-4-3-5. This path is a winning one, and it's easy to calculate the probability that a counter follows this path. Since there are eight arcs in the path and each has probability 1/2 (each arc from any node is equally likely), the probability of this path occurring is $(1/2)^8$. This is not the only way, however, that a player can win. A complicating fact is that there are infinitely many winning paths. To complete the calculation that someone wins \$5 playing this game, we must enumerate all the paths in the graph that start at \$1 and end with \$5, then determine the probability of each path, and finally sum these probabilities.

Table 5 lists some of the paths from \$1 to \$5, together with their lengths and probabilities. One of the two shortest paths always appears at the end of any winning (ending with \$5) sequence of G's and R's. The early part of each path simply shows that you can run through the cycle 1-2-4-3-1 (four light flashes,GGRR) any number of times before you hit on the final part of the path that ends either 1-2-4-5 or 1-2-4-3-5.

So winning paths will have lengths 3, 7, 11, 15, ... or lengths 4, 8, 12, 16, ... Each path of length i has probability $(1/2)^i$. Hence the probability that the counter completes a path that leads from \$1 to \$5 is given by

$$\text{Prob(Winning \$5)} = \sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^{4i+3} + \sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^{4i+4} = \frac{3}{16} \sum_{i=0}^{\infty} \left(\frac{1}{16}\right)^{i} = \frac{1}{5}$$

Sequence of wins/losses	Path in graph	Length of Path	Probability
GGG	1-2-4-5	3	$(1/2)^3$
GGRG	1-2-4-3-5	4	$(1/2)^4$
GGRRGGG	1-2-4-3-1-2-4-5	7	$(1/2)^7$
GGRRGGRG	1-2-4-3-1-2-4-3-5	8	$(1/2)^8$
GGRRGGRRGGG	1-2-4-3-1-2-4-3-1-2-4-5	11	$(1/2)^{11}$
GGRRGGRRGGRG	1-2-4-3-1-2-4-3-1-2-4-3-5	12	$(1/2)^{12}$

Table 5: Some paths from \$1 to \$5 and their probabilities

Exercises

4. Verify by direct calculation (using the graph in Figure 1) that the probability of losing Lotto Game One is 4/5.

5. Suppose each move of the counter in Lotto Game One is no longer governed by red and green flashes that are equally likely. Instead the probability of a green flash is $p = .4$. How does this change the analysis of the problem. Can you find the probability that starting with \$1 a player wins \$5?

6. (**This is an extremely difficult exercise!**) Follow a similar procedure to find the probability of winning Lotto Game Two, assuming that the red and green flashes both occur with probability 1/2:

   (a) First find all shortest paths from \$1 to \$5. Describe them in terms of sequences of red (R) and green (G) flashes.

   (b) Determine what lengths n are possible for winning paths.

   (c) Count the number of winning paths for each possible length n.

   (d) Calculate the probability of winning Lotto Game Two in exactly n flashes.

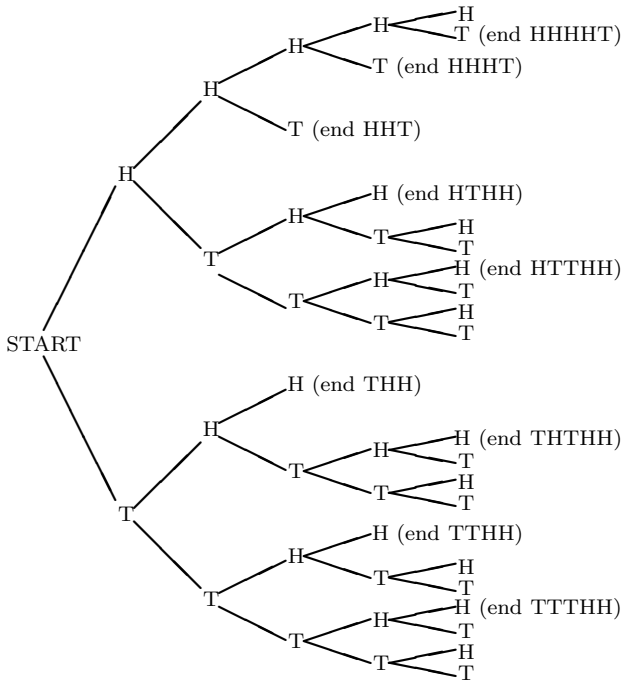
   (e) Calculate the probability of winning Lotto Game Two.

1.4 There's More to Life Than Lotto

The Lotto games in the previous section are played on a game board that looks like a graph: it has nodes and arcs. There are many probability processes that can be represented by such graphs. As an illustration of how infinite probability processes can be analyzed effectively using the graph representation we consider a coin tossing game in which the number of tosses is NOT fixed before the game starts.

Ben and Jerry are going to play a game in which a fair coin is flipped repeatedly until one of the sequences HHT or THH appears. They agree that if HHT comes up first Ben wins and if THH appears first Jerry wins. What is Ben's chance of winning this game? Once again we have a game that could continue forever. For starters, let us look at the beginning of a standard probability tree for this game (Figure 3).

There are several interesting features in this tree. Although there is a fairly obvious repeated structure in the top part of the tree, the repeating sub-trees in the bottom half of the tree are far from obvious. To help gain insight we can take advantage of a directed graph. In this game it is easier to construct the finite graph corresponding to the tree if we focus on the process of recording the outcomes of the coin flips that appear in any string of repeated tosses. First note that the shortest sequence of coin flips that could result in a winner has three flips. The graph showing


START
H
T
H
T
H
T

possible sequences of H's and T's that can occur in the course of the game. To be convinced that the finite graph correctly models this game we need to see that each possible sequence that can occur in a completed game corresponds to exactly one path from START to either HHT or THH. Let's look at an example. Consider the five-flip complete game sequence HTTHH. In Figure 4(b), we can trace the corresponding path: START-H-T-T-TH-THH. Each one of the five coin flips is represented by an edge in this path. The self-loop T-T in the graph corresponds to the fact that repeated tails TT in the sequence of coin flips really doesn't move the game closer to completion. The game only moves closer to an end when heads start to occur after at least one tail. See if you can find the path in the graph that corresponds to the game HTHTTTHT.

Exercises

7. Suppose Ben and Jerry continue playing the coin tossing game but now they agree that if HTT comes up first Ben wins and if TTH appears first Jerry wins. Draw the finite graph diagram that corresponds to this version of the game.

8. Same question as Exercise 7 for the sequences THT for Ben, HTT for Jerry.

9. For each of the five sequences below, determine whether the sequence could appear as a game sequence in the original version of Ben and Jerry's game—HHT against THH (shown in Figure 4(b)). If a sequence doesn't correspond to a possible game sequence for Ben and Jerry explain why.

   (a) HTHTHH
   (b) TTTTTHTHTHH
   (c) HTHTHTHTHTT
   (d) HHHTTHH
   (e) HHHHHHHT

10. Suppose Thelma and Louise decide to play the same type of coin tossing game as Ben and Jerry, but they agree that if HTHT come up Thelma wins and if TTHT comes up Louise wins. Draw the finite graph diagram that corresponds to this game.

11. Can you think of a problem that might arise if the winning strings had different lengths? (Hint: Can you think of an example where Jerry has no chance of winning?)

12. Draw the finite graph diagram that corresponds to this game. Roll a single six-sided die. Your starting score is 0. If you roll an even number, you add that number to your total. If you roll an odd number, you subtract that number from your total never dropping below 0. The game is over when your total reaches at least 7 or it returns to 0.

1.5 Calculating Probabilities Directly

Consider the coin tossing game between Ben and Jerry represented by the finite graph in Figure 4(b). We should be able to calculate the probability that Ben wins by 1) determining how many different paths there are from START to HHT (Ben's sequence), and 2) calculating the probability of each such path. Remember that the probability of any path with i edges is $(1/2)^i$ since each edge represents an outcome for the toss of a fair coin. We see that the only paths from START to HHT follow the top part of the diagram. It is easy to count them. There is exactly one path of length

3, START-H-HH-HHT; exactly one of length 4, START-H-HH-HH-HHT; exactly one of length 5, START-H-HH-HH-HH-HHT; and so on. In total there is one path of length i, for each $i \geq 3$. So

$$\text{Prob(Ben wins)} = \text{Prob(HHT)} = \sum_{i=3}^{\infty} \left(\frac{1}{2}\right)^i = \ldots = 1/4.$$

Of course, we can immediately see that Jerry's probability of winning is $1 - 1/4 = 3/4$. But let's see how to use the graph diagram to calculate the probability of THH directly. This calculation is complicated by two facts. First there are two different direct (i.e., no loops) paths from START to THH: START-T-TH-THH and START-H-T-TH-THH. Second there are two ways to construct longer paths from START to THH, the self-loop at T and the two-cycle T-TH-T. One way to avoid enumerating all paths in this graph is to note that all paths that lead to Jerry winning must contain the node labeled T and that once having arrived at that node, the game can be won only by Jerry. In other words, if the game gets to the node labeled T, the probability that Jerry wins is 1. So all we need to do in this graph is determine the probability that the game gets to node T. This observation substantially simplifies the enumeration of paths. There are only two paths from START to T: START-H-T and START-T. It is a straightforward calculation to determine

$$\text{Prob(Jerry wins)} = \text{Prob(path START-H-T)} + \text{Prob(path START-T)} = \left(\frac{1}{2}\right)\left(\frac{1}{2}\right) + \left(\frac{1}{2}\right) = \left(\frac{3}{4}\right).$$

In the previous example the fact that we quickly reach a single node from which we know with certainty only one player can win allows us to make short work of the probability calculation. This nice feature is, however, often not a feature of the graph we need to work with. Suppose that Ben and Jerry get tired of Jerry winning most of the time and agree to new rules. The game now ends with the first appearance of either string HHT or string THT. Ben still wins with HHT. Before you look at Figure 5, try to draw the finite graph for this game (you can check your work against Figure 5). Notice that in this graph, either player can win from node H and either player can win from node T. There is no node (except a winning one) from which we can be certain that Jerry (THT) wins. A further complication arises from the fact that there are several paths through the graph from START to HH—a node from which only Ben (HHT) can win. The three basic paths (ignoring for the moment the effect of self-loops) from START to HHT are shown in Figure 6.

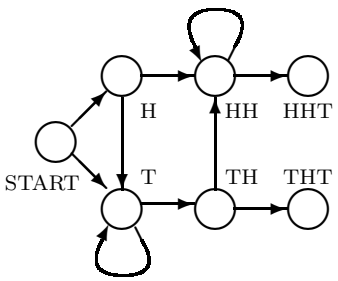


Figure 5: Finite Diagram for New Coin Tossing Game

Each path from START to HHT will be of exactly one of the following forms, corresponding to each of the diagrams in Figure 6: 1) START-H-HH-. . . -HH-HHT, where the substring HH-. . . -HH corresponds to traversing the self-loop any number of times including not at all; 2) START-H-T-T-. . . -T-TH-HH-. . . -HH-HHT, with similar explanations for substrings HH-. . . -HH and T-. . . -T. 3) START-T-T-. . . -T-TH-HH- HH-. . . -HH-HHT, with similar explanations for substrings HH-. . . -HH and T-. . . -T.

Once we have all the paths from START to HHT counted exactly once, we can again find Ben's probability of winning by summing the probabilities of all such paths. It is important to remember that each path of the form START-H-HH-HH-...-HH-HHT has a different probability depending on the number of times each self-loop is traversed: each time a self-loop is traversed the path's probability is multiplied by (1/2). The calculation can be summarized as

$$\begin{aligned}
\text{Prob(Ben wins)} &= \text{Prob(HHT)} \\
&= \left(\frac{1}{2}\right)^3 \left(\sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^i\right) + \left(\frac{1}{2}\right)^5 \left(\sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^i\right) \left(\sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^i\right) \\
&\quad + \left(\frac{1}{2}\right)^4 \left(\sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^i\right) \left(\sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^i\right) \\
&= 5/8
\end{aligned}$$

The preceding enumeration of paths and calculation of path probabilities is certainly more complicated than the calculation in Lotto Game One, but it is much easier than many of the calculations that one may encounter in such games. In fact, any time there are self-loops or cycles in the diagram, it can be quite difficult to carry out a straightforward calculation. Don't despair. In the next section we'll develop an algebraic method of exploiting the finite graph structure to determine the probability of winning any coin tossing game like Ben and Jerry's.

Exercises

13. Consider the coin-tossing game discussed above in which two players bet on the strings HHT and THT. We already computed the probability that HHT wins by enumerating the paths from START to HHT. That probability was found to be 5/8. Given this information, we already know that the probability of THT winning is $1-5/8 = 3/8$. Verify this by enumerating the paths from START to THT and computing the probability directly.

14. Draw the finite graph for the coin tossing game in which two players bet on the strings HTH and THT. In theory, it is possible to calculate the probability that HTH wins directly from an analysis of all paths that end at HTH, as in the example given in the text above. In practice, however, it is rather difficult to carry out such an analysis for this game (Why?). Can you think of another way to calculate the probability that HTH wins this game?

15. Draw the finite graph for the coin tossing game in which two players bet on the strings HHTH and THHT. Comment on the feasibility of enumerating all paths from START to HHTH and calculating the their probabilities.

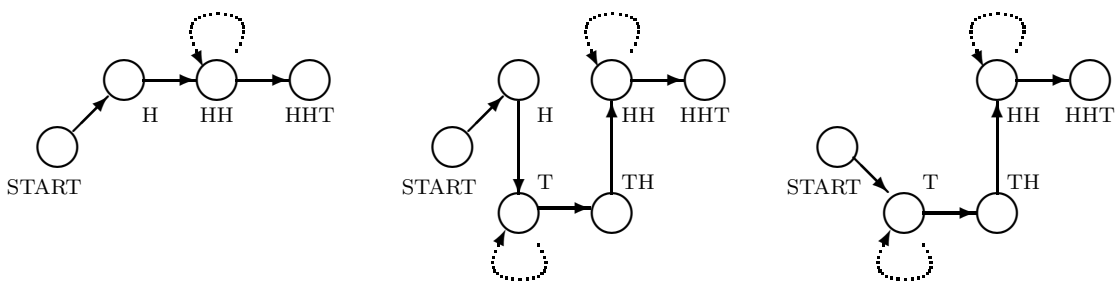


Figure 6: Possible paths from start to HHT

16. Suppose three people want to place a coin tossing game in which A wins if HHT appears first, B wins if THT appears first, and C wins if TTT appears first. Can this game be represented by a finite graph in a way analogous to the way two-person games have been?

2 Backtracking

2.1 Computing Probabilities by Backtracking

Suppose we wish to compute the probability that the sequence HHT will defeat the sequence THH in the coin tossing game. Let a be the probability that HHT wins. Then $1 - a$ is the probability that THH wins.

Think of probability as a fluid mass flowing at a constant rate through the system described by the diagram in Figure 4(b) we have already produced for this game. We will introduce probability mass into the system through the starting node at a constant rate of 1 and observe the rate at which the mass flows through the each of the nodes in the system. The probability that the sequence HHT wins the game will be the rate at which probability mass flows out of the system into node HHT. Similarly, the probability that sequence THH wins the game will be the rate at which probability mass flows out of the system into node THH. We will label each node and arc with the rate of flow of probability mass passing through it. We begin by placing a 1 inside the starting node to represent the constant flow rate of 1 into the system. Nodes HHT and THH are labeled a and $1-a$ respectively (see Figure 7). Throughout this process, we need only make sure that the flow rate into each node is the same as the flow rate out of that node. That is, we want a "conservation" equation to hold.

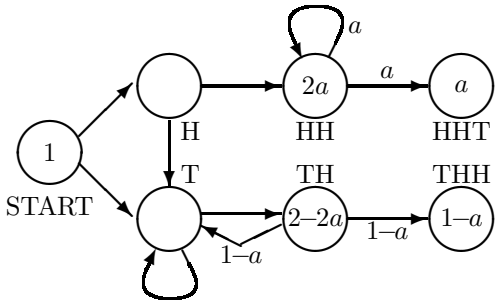


Figure 7: The first few steps of the backtracking process

Analysis for Nodes HHT and THH

Because probability is flowing into node HHT at rate a, and because the only arc pointing to node HHT comes from node HH, the flow rate along that arc must be a. Now we need to be a bit careful. There are two arcs leaving node HH. Those arcs represent the paths taken when either a head or a tail comes up on the next toss. Those two outcomes are equally likely, so the probability of following either of the arcs must be equal. Since we know that the probability of following the arc from node HH to node HHT is a (remember, probability is being identified with the rate of flow through the system), the probability of following the loop from node HH to itself must also be a. Since those two arcs are the only ones leaving node HH, the mass flowing through node HH must be moving at a rate of $2a$. Therefore, we will write $2a$ inside node HH (see Figure 7).

By similar arguments, we can label the arc from node TH to node THH with $1 - a$, the arc from node TH to node T with $1 - a$, and label node TH with the total mass flowing through that

node, $2-2a$ (again, see Figure 7). We will continue to work backwards through the flow chart, always being careful to balance the flow rate into a node with the flow rate out of the node.

Analysis for Nodes HH **and** H

There are two arcs pointing into node HH, one from node H and the loop at node HH. We already know that the mass is flowing through node HH at a rate of $2a$, and that the flow rate through the loop is a (think of the loop as "recycling" probability mass back into the node). Since the loop is recycling mass at a rate of a, the arc from node H to node HH must be carrying probability mass at a rate of $2a - a = a$ as well. This allows us to conclude that the flow rate along the arc from node H to node T is also a, as the two arcs leaving node H must have equal probability. We complete the upper part of the diagram by noting that the only arc into node H is the one from the starting node. In order to balance the flow rate of $2a$ out of node H we must have a flow rate of $2a$ from the starting node to node H (see Figure 8).

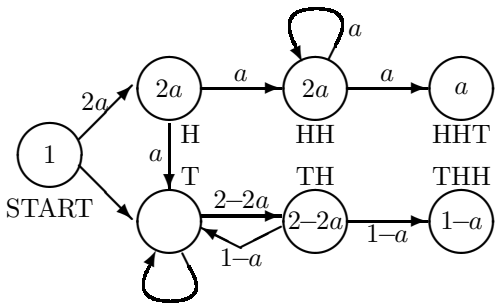


Figure 8: The backtracking process at Node H

Analysis for Nodes TH **and** T

We already know that the flow rate out of node TH is $2-2a$. The only arc pointing into node TH is the one from node T, so that arc must carry the entire flow rate of $2-2a$ (see Figure 8).

Because there are two arcs pointing out of node T, each must have the same flow rate. So the loop from node T to itself must have a flow rate of $2-2a$ (as before, think of a loop as carrying recycled mass) giving a total rate of $4-4a$ flowing through node T. That means we need a total rate of $4-4a$ flowing into node T. We already know about some of that flowing mass: the loop at node T is recycling mass at a rate of $2-2a$, the arc from node H to node T carries mass at a rate of a, and the arc from node TH to node T carries a flow rate of $1-a$. So the arc from the starting node to node T must carry the remaining mass at a rate of $(4-4a)-[(2-2a)+a+(1-a)] = 1-2a$. This completes the labeling of our flowchart (see Figure 9).

We can finally determine the value of a. Because each arc out of a given node represents tossing either a head or a tail on the next toss, each arc out of a node must carry half of the probability mass leaving that node. Recall that we started by putting probability mass into the starting node for our system at a rate of 1. Thus each arc out of node T must have a flow rate of $1/2$. On the other hand, we have determined that the arc from the starting node to node H carries mass $2a$ and the arc from the starting node to node T carries a mass of $1-2a$. This gives us two linear equations in a: $2a = 1/2$ and $1-2a = 1/2$. Regardless of which equation we choose to solve we get the solution $a = 1/4$. So the sequence HHT has one chance in four to beat the sequence THH in our game. Note that we did not need to label every arc and node in order to arrive at the final solution. As soon as we knew that the arc from the starting node to node H was labeled $2a$,

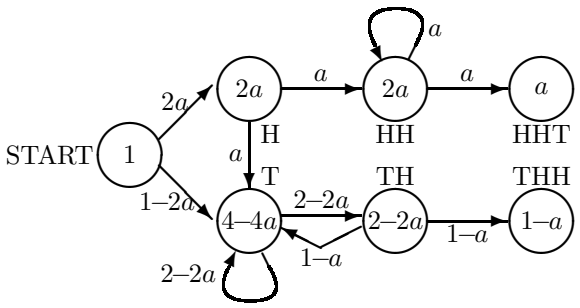


Figure 9: The backtracking process at Node T

we could have concluded that $2a = 1/2$, and therefore that $a = 1/4$. However labeling the entire flow chart will prove useful for some other calculations, as we shall soon see. In fact, now that we have a value for a, it makes sense to re-label our entire flow chart with numerical values (see Figure 10).

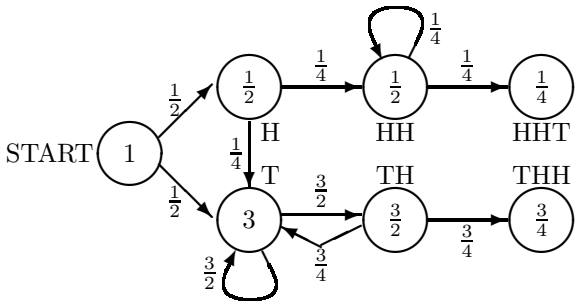


Figure 10: The final backtracking diagram after solving for a

Exercises

1. In Section 1 we presented a new coin tossing game in which Ben wins if the sequence HHT appears first and Jerry wins if the sequence THT appears first (see Figure 5, Page 11). Using infinite series we were able to calculate that Ben's probability of winning this new game is 5/8. Confirm this result using backtracking.

2. Try to use the backtracking technique described above to find the probability that you will lose in the Lotto Game Two example.

3. For each box in the table, find the probability that the row sequence beats the column sequence. Is it possible to determine some of the probabilities by using complement and symmetry arguments rather than by backtracking?

	HH	HT	TH	TT
HH	...			
HT		...		
TH			...	
TT				...

4. Use the backtracking to determine the probability that the sequence HHTH beats the sequence THHT. You were asked to create the diagram for this game in Exercise 15 of Section 1.

5. Suppose that we play the coin tossing game with a biased coin. If the probability of getting a head on any given toss is 2/5, determine the probability of sequence HHT beating sequence THH.

2.2 Using Backtracking Results to Compute Expected Values

We now have a method for finding the probability that a given sequence will win the coin tossing game. Perhaps we can find out how long such a game can be expected to last. In other words, we will try to determine the average number of tosses that are required to complete a coin tossing game. We will restrict our discussion to three-toss sequences so that we can refer to the examples discussed above. However, the methods we discover can be applied to games involving sequences of any finite length. Consider the example above in which we pitted the sequence HHT against the sequence THH. It is easy to see that the game could end in three tosses (the first three tosses could be one of the two target sequences). It may also be possible to see sequences like HTTHTTTHTHTHTTHTHTTHTHTHTTHTHTHTTTHTTHTHTHH, in which the sequence THH wins, but only after 43 tosses! (Do you think such a game is possible? Do you think such a game is likely?) So what should we expect as an "average" number of tosses before a winner is determined? We call this number the expected value of the number of tosses (or simply the expected number of tosses). We can estimate the expected number of tosses by actually playing the game several times and finding the average number of tosses per game.

Activity 2.2

> Have students work alone or in pairs. Each student/pair will toss a coin and record the result of each toss. When either of the two sequences HHT or THH appears as the last three tosses, stop the game and record the number of tosses it took to end the game. Repeat this process a total of five times for each student/pair, and determine the average number of tosses per game. That number should be near the expected number of tosses.

The results of Activity 2.2 will be fine as an approximation to the expected number of tosses, but how can we determine the *exact* expected number of tosses? Before we attempt to do so, we'll give a brief review of the usual method for computing expected value for finite probability distributions.

Expected Value

What exactly do we mean by expected value? Consider a simple game of chance in which players bet on the outcome of a spinner. A spinner is a wheel with several numbered slots. For this example assume that there are 100 slots on the spinner and that players may bet \$1 on the outcome. If the player chooses correctly, then he wins \$20. Otherwise, he loses the \$1 he bet. In this example a player can either have a net gain of \$19 or a net gain of −\$1 (we think of a net loss as a negative net gain for computational purposes). In any single instance of game, a player can win \$19 or −\$1. However, if the player plays the game many times the frequency of the outcomes\$19 or −\$1 will determine the "average" amount won or lost. So the player "expects" this "average" return for his \$1 bet game is played many times.

Suppose that a person played this game many, many times. In the long run, that person would win approximately one out of every 100 games and lose approximately 99 out of every 100 games. Then that player would win \$19 about 1/100 of the times she plays and lose \$1 about 99/100 of

the times she plays. So in the long run, she expects an outcome of $(\$19)(1/100) + (-\$1)(99/100) = \$19/100 - \$99/100 = -\$80/100$, or $-\$.80$ for every dollar bet. In other words, for every dollar that she bets, she *expects* to lose 80 cents. How did we compute her expected winnings (here we regard her loss of 80 cents as *negative* winnings)? We took each possible outcome of the game and multiplied it by the probability of the game resulting in that outcome. Then we summed all of those products. Because the outcome of the game depends on probabilities, it can be represented by a random variable X, whose possible values are $X_1 = \$19$ and $X_2 = -\$1$.

We can generalize this procedure to other random variables. Let X be a random variable that can take on any of the values $X_1, X_2, \ldots$ (for our purposes, it may be necessary to let the list of possible outcomes be finite or countably infinite). Let $\text{Prob}(X_i)$ represent the probability that X_i is the value taken by the random variable X. Then the expected value of the random variable X is defined by $\sum X_i \text{Prob}(X_i)$, where the sum given is either finite or infinite, depending on whether X takes on finitely or countably many values.

In our coin tossing game, we can let X be the random variable representing the number of tosses necessary to determine a winner. Note that X can take on any integer value greater than or equal to 3. In other words, $X_i = i$ for $i \geq 3$. So to compute the expected number of tosses we would have to evaluate the infinite sum $\sum_{i=3}^{\infty} i\text{Prob}(X_i)$. While it is correct to compute the expected number of tosses in this way, there are some difficulties. First, we must determine $\text{Prob}(X_i)$ for every integer $i \geq 3$. It is not at all clear that this will be possible. Second, even if we can compute $\text{Prob}(X_i)$ for all $i \geq 3$, we must then be able to determine the value of the infinite sum. This is not always possible.

Let's see if we can formulate a reasonable way to use our flowchart to determine the expected number of tosses required to complete the coin tossing game with sequences HHT and THH. The expected number of tosses in a game is exactly the same as the expected number of times that we visit any node as we move through the flowchart during a game. That's because each visit to a node represents the result of another toss of the coin. So in order to determine the expected number of visits to nodes in the flowchart, we can try to find the expected number of visits to each *particular* node, and then add up those expected values.

In our coin tossing game, each time we traverse an arc we visit another node. So the expected number of visits to a node is given by the sum of the flow rates along all arcs pointing into that node! If we refer to Figure 10 we can easily compute the expected number of visits to each node (see Table 6).

Node	Expected Visits
H	1/2
T	3
HH	1/2
TH	3/2
HHT	1/4
THT	3/4

Table 6: Expected number of visits to each node in the Coin Tossing Game

Because each visit to a node is the result of a single toss of the coin, the expected number of visits to a given node is equal to the expected number of times that a toss of the coin will bring us to that node. Since a single toss of the coin can and must result in exactly one visit to a node, the total number of coin tosses that we expect to use in completing this game is equal to the sum of the expected visits to all of the nodes in the game. As we can see from Table 6, that sum is 13/2,

or 6.5. In other words, if we were to play this game repeatedly and record the number of tosses used for each repetition of the game, then we should average about 6.5 tosses per game. How does this compare to the empirical data collected in Activity 2.2?

Exercises

6. In Exercise 2 you found the probability of losing Lotto Game Two. What is the expected amount of money that you will have if you play Lotto Game Two to its conclusion?

7. Find the expected number of tosses needed to complete the coin tossing game between the sequence HH and TH.

8. In Exercise 4 you determined the probability that the sequence HHTH beats the sequence THHT in the coin tossing game. Find the expected number of tosses in a game between these two sequences.

9. In Exercise 5 you found the probability of sequence HHT beating sequence if the probability of getting a head on any given toss is 2/5. Is the expected number of tosses the same as it was when we used a fair coin?

3 Calculating Probabilities Using the Addition and Multiplication Rules (Optional)

Note to the instructor: *This section describes an alternate approach to computing the probability of winning the coin tossing game described in the first section. You can choose to replace the previous method with this one, or include both methods. This section requires some additional knowledge of probability theory, in particular, the concept of "conditional"probability.*

Consider the first coin tossing game introduced in Section 1 and analyzed in Sections 1 and 2. That game was between sequences HHT and THH and is described by the diagram in Figure 11.

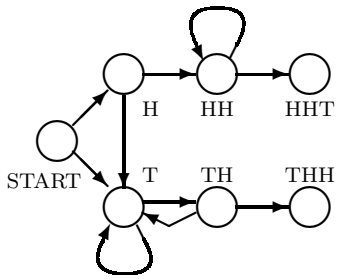


Figure 11: Diagram for the Coin Tossing Game

We wish to compute the probability that the sequence HHT will beat the sequence THH in our coin tossing game. If you have completed Section 2 then you have already calculated this probability and will be able to compare that result with the one given here. In any case, in this section we will be describing an alternate approach to finding this probability.

Let us change the problem slightly. Instead of thinking about the probability that HHT beats THH from the beginning of the game, let us consider the problem of finding the probability that

HHT beats THH *given* that you are already on a particular node in the graph, such as node H. We will denote this probability by Prob(HHT|H).

We will take care of the easy cases first. Suppose that we are already on node HHT. Then of course we are done since HHT has already won. So

$$\text{Prob}(\text{HHT}|\text{HHT}) = 1.$$

Now suppose that we are on node THH . Then THH has won and

$$\text{Prob}(\text{HHT}|\text{THH}) = 0.$$

To find the probabilities at the remaining nodes, we will need two basic rules for probability: If A and B are mutually exclusive events, then

$$\begin{aligned} \text{Prob}(A \text{ or } B) &= \text{Prob}(A) + \text{Prob}(B) && \text{(Addition Rule for Mutually Exclusive Events)} \\ \text{Prob}(A \text{ and } B) &= \text{Prob}(A)\text{Prob}(B) && \text{(Multiplication Rule)} \end{aligned}$$

So to start, suppose that you are on node HH. Looking at Figure 11, we see that there are two ways that HHT could win from HH:

1. the next flip yields an H, and HHT wins from HHH = HH; or
2. the next flip yields a T, and HHT wins from HHT.

Let us abbreviate the phrase "the next flip is H" by nf = H. Then, using the addition and multiplication rules for probability,

$$\begin{aligned} \text{Prob}(\text{HHT}|\text{HH}) &= \text{Prob}(nf = \text{H} \textbf{ and } \text{HHT}|\text{HH}) + \text{Prob}(nf = \text{T} \textbf{ and } \text{HHT}|\text{HHT}) \\ &= \text{Prob}(nf = \text{H})\text{Prob}(\text{HHT}|\text{HH}) + \text{Prob}(nf = \text{T})\text{Prob}(\text{HHT}|\text{HHT}) \end{aligned}$$

We are assuming that we have a fair die, so

$$\text{Prob}(nf = \text{H}) = \text{Prob}(nf = \text{T}) = 1/2,$$

So if we set Prob(HHT|HH) = a, we get the following equation:

$$\begin{aligned} a &= (1/2)a + (1/2)(1) \\ (1/2)a &= 1/2 \\ a &= 1 \end{aligned}$$

So, Prob(HHT|HH) = 1. This makes sense since once you are at HH, HHT must win eventually.

Now we have to choose the next node to work on. Notice that the probability at node X depends on the probabilities of getting to each successor node Y and on the probability of HHT winning from Y. It makes sense then to identify nodes that have no successors before we begin our analysis, for we must begin our analysis there. A node that has no successors will never fire any chips, and for that reason is often referred to as a "sink node." In hindsight, we chose node HH first since it depended only upon the sink node HHT (we knew Prob(HHT|HHT) = 1) and itself (a self-loop comes out of node HH). Each of the remaining nodes requires knowledge of at least one interior node. So let's choose the node that is successor to as many remaining nodes as possible.

That would be node T. Let Prob(HHT|T) = b. A cursory examination of Figure 11 tells us that $b = 0$. However, let us confirm that with a calculation. Now

$$\begin{aligned}\text{Prob(HHT|TH)} &= \text{Prob}(nf = \text{H})\text{Prob(HHT|THH)} + \text{Prob}(nf = \text{T})\text{Prob(HHT|T)} \\ &= (1/2)(0) + (1/2)b \\ &= (1/2)b\end{aligned}$$

Now

$$\begin{aligned}\text{Prob(HHT|H)} &= \text{Prob}(nf = \text{H})\text{Prob(HHT|HH)} + \text{Prob}(nf = \text{T})\text{Prob(HHT|T)} \\ &= (1/2)(1) + (1/2)b \\ &= (1/2) + (1/2)b\end{aligned}$$

Finally, we have

$$\begin{aligned}\text{Prob(HHT|START)} &= \text{Prob}(nf = \text{H})\text{Prob(HHT|H)} + \text{Prob}(nf = \text{T})\text{Prob(HHT|T)} \\ &= (1/2)((1/2) + (1/2)b) + (1/2)b \\ &= (1/4) + (1/4)b + (1/2)b \\ &= (1/4) + (3/4)b\end{aligned}$$

Filling in each of the nodes of the diagrams with the probability that HHT wins from that node, we have Figure 12(a).

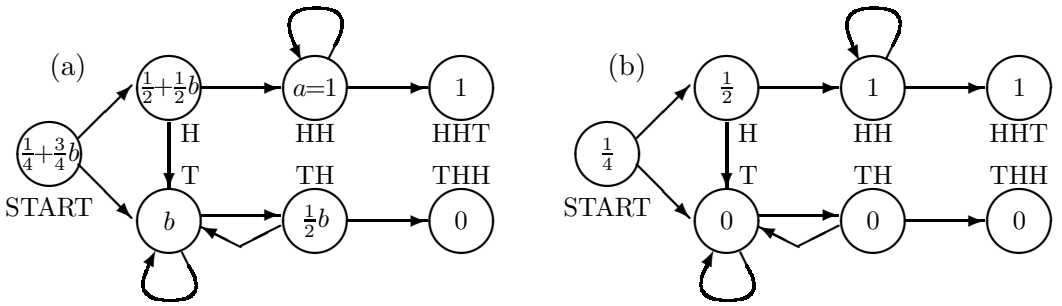


Figure 12: Probability that HHT wins from each node

All that remains is to find the value of Prob(HHT|T). We note that

$$\text{Prob(HHT|T)} = \text{Prob}(nf = \text{H})\text{Prob(HHT|TH)} + \text{Prob}(nf = \text{T})\text{Prob(HHT|T)}$$

and so

$$\begin{aligned} b &= (1/2)(1/2)b + (1/2)b \\ b &= (3/4)b \\ b &= 0 \end{aligned}$$

The completed diagram is given in Figure 12(b). There are several observations to make.

1. We could have determined the value of b right from the start by noting that there are no paths from T to HHT .

2. Prob(HHT|start) $= \frac{1}{4}$ is the probability that HHT will win the game. Those who have completed Sections 1 and 2 will notice that this result agrees with the results obtained for this game in each of those sections.

3. There are cases where it is necessary to use one or two variables to represent the probabilities at the nodes, and find values for them by solving simultaneous equations.

Exercises

1. Consider the game HHT vs. THT played with a fair coin.

   (a) Draw the graph representing the game.

   (b) For each node X, find Prob(HHT|X).

2. Consider the game HHT vs. THT that was discussed above. Suppose that you are using a biased coin with Prob(nf = H) = 1/3. For each node X, find Prob(HHT|X).

4 Chip Firing Games

4.1 What is a Chip Firing Game?

The problems that we solved in the previous sections required that we either use infinite series or the method of backtracking. Later in this section we will see that there is a way to use the diagrams we constructed to solve probability problems in a discrete finite way. First, we will learn a very simple game that can be played on the diagrams that we have been constructing. *Chip firing games* are games played on directed graphs (or networks). We can think of chips as poker chips, checkers, markers, or any other small discrete object. Chips are placed on the nodes of a graph (we may place chips one at a time or more than one at a time). When the number of chips on a node is greater than or equal to the number of arrows leaving that node, the node *fires*, sending one chip along each of the arrows out of the node. If a node has no arrows leaving it, then it is impossible for that node to fire. We will call such nodes *boundary nodes*. Note that boundary nodes absorb all of the chips that are fired into them. For that reason, boundary nodes are sometimes called *absorbing nodes*. Nodes that are not boundary nodes will be called *interior nodes*. We say that an interior node is *critically loaded* when it contains one fewer chip than would be required for the node to fire. Thus, adding one or more chips to a critically loaded node will cause it to fire. We say that the chip firing game is in a *stable configuration* if none of the interior nodes are in the process of firing. The two extreme examples of stable configurations are the *empty configuration*, in which none of the interior nodes contain any chips, and the *critically loaded configuration*, in which every interior node is critically loaded. There are no rules about what the starting configuration must look like, except that it be stable. We can begin with the empty configuration, the critically loaded configuration, or any other stable configuration. In Figure 13 we have an example of a chip firing game. The directed graph for this game is the graph for Lotto Game One which we introduced in the Introduction (see page 4). Figure 13 (a) shows the chip firing game in the empty configuration. The \$0 and \$5 nodes are boundary nodes and the \$1, \$2, \$3, and \$4 nodes are interior nodes. Figure 13 (b) shows the game in a stable configuration that is neither the empty configuration nor the critically loaded configuration. Figure 13 (c) shows the game in the critically loaded configuration—if another chip is added to any interior node, that node must fire.

We should make a few comments about what can happen once nodes begin firing in a chip firing game. Suppose that the directed graph we use in a chip firing game is a cycle of any length

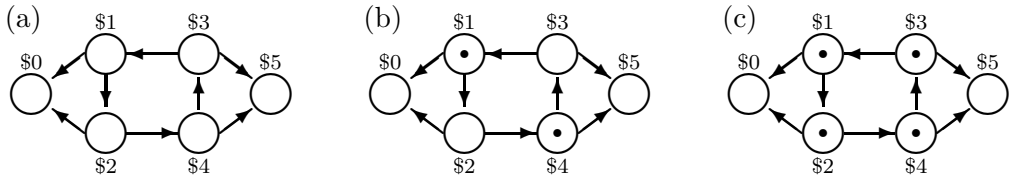


Figure 13: An Example of a Chip Firing Game

in the empty configuration. If we drop a single chip on any node, then that node will fire sending the chip to the next node in the cycle which will then also fire in turn. This single chip will be fired from node to node around the cycle forever, and so the game will never end.

Suppose on the other hand that the directed graph we use has the property that from every node in the graph there is a path to a boundary node. The diagrams for the lotto games in the Introduction and the diagrams of the coin tossing games in Sections 1, 2, and 3 are examples of such graphs. In such a graph, when a node fires, at least one chip is sent along a path leading to a boundary node. Suppose a node were to fire repeatedly because it contained a large number of chips. Eventually all of its neighbor nodes would begin to fire, and then those nodes' neighbors would fire, and so on. Because every node is on a path leading to a boundary node, chips would eventually have to be absorbed by the boundary nodes thus reducing the number of chips available to fire. So if the number of chips in the game is finite, the nodes must all stop firing eventually.

4.2 Tossing Coins and Firing Chips

Let's return to the coin tossing game we introduced in Section 2. The diagram we constructed is an example of a finite directed graph (see Figure 14(a)) meaning that there are a finite number of nodes connected by arrows indicating the direction of movement. We will play the chip firing game on this graph, starting with the critically loaded configuration. (Note: You may want to draw an enlarged sketch of the graph and find some chips so that you can follow along with the game yourself.)

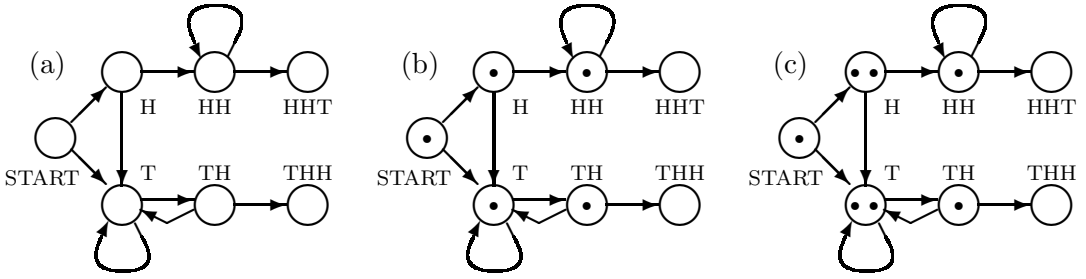


Figure 14: The Coin Tossing Game is now a Chip Firing Game

Because each of the interior nodes has two arrows leaving it, the critically loaded state consists of a single chip on each interior node. We will leave the boundary nodes empty. Our graph now looks like Figure 14(b). We are going to add chips to the START node and see what happens.

Note that it doesn't matter whether we add chips to the START node one at a time or two at a time. If we add only one chip, the START node will fire leaving no chips there. If we then add another single chip, there will be only one chip on the START node and it will not fire. If at the beginning we add two chips to the START node simultaneously, that node would fire leaving one chip. Thus, adding chips one at a time and adding chips two at time yield the same result—the

START node fires once and there is one chip left on it. Here, we will add chips two at a time to speed up the process. As an exercise, you can try the analysis we are about to embark upon by adding chips one at a time to verify that the final result will be the same.

If we place two chips on the START node it will fire, sending one chip to node H and one chip to node T, giving us the configuration in Figure 14(c).

Now node H and node T each contain two chips, so both of them need to fire. It is difficult to keep track of the chips when we try to fire several nodes simultaneously, so we will fire them one at a time. An important question to consider: Does the order in which we fire the nodes matter? In other words, will we end up with the same result regardless of which node we fire first? Let's worry about that later. At this point we will assume that it doesn't make any difference and that we can fire the nodes in any order. If we fire node H, then one chip will go to node HH (putting a total of two on that node) and one chip will go to node T (giving us a total of three chips on that node). Node H is now done firing so we can move to another node. Note that we still have two nodes that need to be fired, node HH and node T. If our assumption that the order doesn't matter is valid, then it still shouldn't make any difference which one we fire first. Let's fire node T. Even though node T contains three chips, only two of them will fire. (Remember: when a node fires, it sends one chip along each arrow out of the node.) So when node T fires, it will send one chip to node TH and one chip around the loop back into node T. Since one of the chips in node T never moved, we are left with two chips in node T. We have to fire node T again! And node HH is still waiting! As you can see, sometimes just a few chips can lead to a big chain reaction, or avalanche, of firings. Anyway, if we fire node T again, one more chip will go to node TH and one will pass along the loop back into node T. Node T has finally finished firing. If you have been playing along, you should now have the (unstable) configuration in Figure 15.

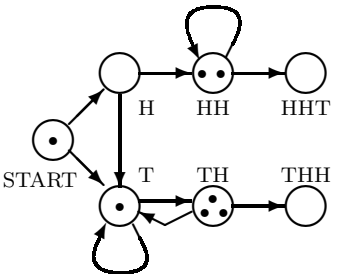


Figure 15: Nodes HH and TH must still fire

Node HH has been waiting a long time to be fired, so we will fire that one next. When we do, one chip will go to node HHT and the other will follow the loop back to node HH. Now node HH has only one chip and therefore no longer needs to be fired. At the moment, only node TH needs to be fired, as it has three chips. When we fire node TH, one chip fires into node THH and another fires into node T, leaving one chip in node TH. So node TH has finished firing. However, the chip that fired into node T has been added to the one that was already there. So node T must be fired again! (see Figure 16(a)).

As before, when node T fires, it sends one chip to node TH (for a total of two—another firing necessary) and one back to itself. So node T now has one chip and node TH has two. When we fire node TH, one chip gets sent to node THH (which now has two) and the other gets sent to node T (which yet *again* has two!). However, node TH is now empty. So when we fire node T (for the last time, we hope), one chip goes to node TH and the other goes back into node T, each of which now contains one chip. We have finally reached the stable configuration shown in Figure 16(b). Adding those two chips to the system forced us to make 18 moves before we reached the stable

configuration (a *move* can be defined as the passing of one chip along one arrow, including loops).

Question 1: For chip firing games in general, do you think that it is possible to keep adding chips to the START node and never see any stable configurations that repeat themselves eventually? Under what circumstances might this happen?

Question 2: For chip firing games in general, if we keep adding chips to the START node must the critically loaded configuration eventually appear? What if we *start* the game in the critically loaded configuration? Must it *reappear*?

Question 3: In the example above, do you think we could have ended up with a different stable configuration if we had fired the nodes in a different order when we had a choice?

Let's keep going from the configuration in Figure 16(b) (Try to follow along and see if you arrive at the stable configuration shown in Figure 17). Putting two more chips on the START node will cause it to fire one chip to node H and one chip to node T. Now each interior node has one chip except for node T, which has two. When node T fires, one chip is returned to node T and one chip goes to node TH. Now node TH is the only node with more than one chip, so it fires next, sending one chip to node THH and one back to node T. Node TH is now empty and node T has two chips again. So we fire node T. This puts one chip in node TH and one back in node T. At this point, which took another 8 moves to reach, the game has returned to the critically loaded configuration (each of the interior nodes has one chip—see Figure 17). We will call a configuration in a chip firing game a *recurrent configuration* if it repeats itself after some number of chips have been added to the system. In the particular game we have been playing, the critically loaded configuration is a recurrent configuration.

4.3 Chip Firing and Probability

There are a few observations that we can make. First, we had to add four chips to the game in order to return to the configuration we started with (critically loaded). Second, adding those four chips caused a total of four chips to exit the system into the boundary nodes HHT and THH, with 1/4 of them in node HHT and 3/4 of them in node THH (those fractions should ring a bell). Finally, a careful count of the moves we had to make until we returned to the starting configuration gives us a total of 26. So on average, each of the four chips added to the system contributed $26/4 = 6.5$ moves to the game (that bell should be ringing again).

Something unexpected has happened. By playing the chip firing game with our diagram, we have managed to stumble upon probabilities and expected values that we had computed in Sections 2 and 3:

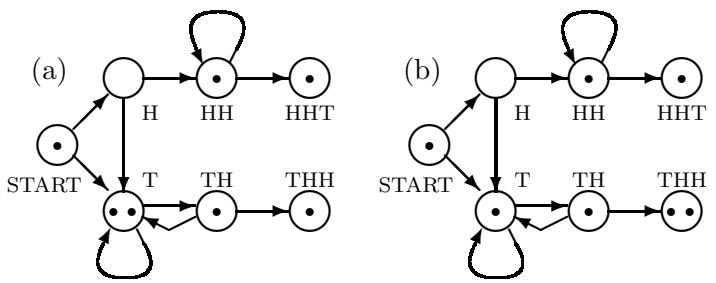


Figure 16: Now Node T must fire

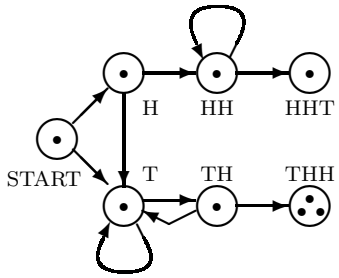


Figure 17: Back to the Critically Loaded Configuration

- the probability of sequence HHT beating THH (1/4 = the proportion of chips that ended up in node HHT)
- the probability of THH beating HHT (3/4 = the proportion of chips that ended up in node THH)
- the expected number of tosses needed to complete a game between those two sequences (6.5 = the number of moves divided by the number of chips added to the system, which after firing returned the game to the starting configuration)

Exercises

1. Verify Exercise 1 from Section 2 with a chip firing analysis.
2. Verify Exercise 2 from Section 2 with a chip firing analysis.
3. Verify Exercise 3 from Section 2 with a chip firing analysis.
4. Verify Exercise 7 from Section 2 with a chip firing analysis.
5. In this exercise we will try to construct a chip firing model that can compute probabilities for a coin tossing game in which the coin is not fair.
   (a) Draw a diagram for the chip firing game between sequences HHT and THH in the case where the probability of tossing a head is 2/5 (This is the game introduced in Exercise 5 of Section 2).
   (b) (This exercise is a true test of patience) Use the model you constructed in part (a) to verify the probability and expected value you calculated in Exercises 5 and 9 (respectively) of Section 2. [Note: You must still follow the chip firing rules as they are given at the beginning of this section.]
6. Repeat your analysis of the chip firing games in Exercises 1, 2, 3, and 4. This time, when faced with a choice of which node to fire next, choose differently than you did when you first completed those exercises. Compare your final results to the results you obtained originally.

4.4 From Chips to Crumbs to Sand to Dust

There is a logical question to ask at this point: What does this chip firing game have to do with probabilities? It hardly seems likely that this could be a coincidence. Well, it isn't, and we will spend some time proving it in Section 5. First, we conclude this section with an *intuitive* idea of why the chip firing game yields the same probabilities that we found in Sections 2 and 3.

In Section 2 we used the notion of probability mass flowing through a system in order to justify the calculations that we performed. Half of the mass flowing out of each node flowed through each of the two arrows pointing out of that node. Here, instead of thinking of probability mass as a fluid, we can think of it as discrete chips. The firing of chips from a node (with one chip passing along each arrow out of that node) corresponds to one unit of probability mass passing through that node.

Suppose that we break each chip into n small pieces (crumbs) and multiply the number of arrows between each pair of nodes by n. Then an interior node in the graph will fire when $2n$ chips are on that node, with n crumbs going to each of the two nodes that originally received whole chips from that (firing) node. So the total amount of "discrete probability mass" firing from node to node is the same as in the chip firing game described earlier in this section.

Now let n get larger and larger. As our chips are broken into smaller and smaller pieces it becomes easier to think of them as grains of sand. Eventually, as n gets very large, we can think of our original whole chips as having been broken into pieces that are so small that the mass resembles small piles of powder or dust. In such a form it is easy to imagine the mass *flowing* from node to node rather than firing in discrete steps. In other words, the discrete chips in the chip firing game represent the same probability mass that we described flowing through the graph in Section 2, just in a different form.

5 Coin Tossing and Chip Firing as Markov Processes

Now that we have seen how the finite chip firing game appears to model the essentially infinite coin tossing game several questions spring to mind: Does it always work for coin tossing? Can chip firing models be used to find probabilities in other situations? What features must these other situations have in order to be good candidates for chip firing models? Most fundamental of all: How does counting a few chips give exact probabilities for games that can be arbitrarily long? To answer these questions we need to achieve a more complete and precise understanding of exactly how the chip firing method of calculating probabilities for coin tossing games works. First let's be clear about what a coin tossing game is. Each person in a group of players chooses a different string of Hs and Ts on which he wants to bet. A coin is tossed repeatedly. The player whose string appears first is the winner, and the game ends. In this general description of coin tossing games, we are not assuming a fair coin is used. Further, in the general game, we allow players to pick strings of different lengths, and we let more than two players compete. In previous sections, we have seen how a coin tossing game can be represented on a finite graph where the nodes correspond to the pertinent ends of strings, while directed edges indicate that another toss has been made. We saw also that firing chips through a graph diagram and counting the chips that landed in the boundary nodes gave exact probabilities for winning. This is the chip firing model of coin tossing games that we want to formalize and to analyze.

5.1 From Chip Firing to Transition Probabilities

As a first step toward this analysis, we need some useful general notation. Observe that each of the nodes in our chip firing graph falls into one two categories: *interior nodes* which have arrows pointing out (allowing chips to pass out of them) and *boundary nodes* which only have nodes pointing in (and therefore absorb all chips that enter them). Given any chip firing diagram, we will let I denote the set of all interior nodes and B the set of all boundary nodes. The individual nodes in the graph will be denoted by s_i for $i = 1 \ldots n$, where n is the total number of nodes in the graph. In order to simplify matters, we will let $I = \{s_1, s_2, \ldots, s_r\}$ and $B = \{s_{r+1}, s_{r+2}, \ldots, s_n\}$.

Because in a general setting we must allow for the possibility of chips being fired between any pair of nodes it is reasonable to organize some information in a matrix. In particular, we can construct a matrix A that has as its ij entry the number of chips that pass from s_i to s_j each time that node s_i fires. In the example of the previous section we used the coin tossing game between sequences HHT and THH to demonstrate how a chip firing game works. Let us construct a matrix for that example. Here we will let $s_1 =$ START, $s_2 =$ H, $s_3 =$ T, $s_4 =$ HH, $s_5 =$ TH, $s_6 =$ HHT, and $s_7 =$ THH. In this case, we end up with the following matrix (the horizontal and vertical lines are used to separate the interior nodes from the boundary nodes):

$$A = \left(\begin{array}{ccccc|cc} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ - & - & - & - & - & - & - \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right) \tag{1}$$

There are a few problems with this approach however. There are two types of possible entries regarding the number of chips that leave a boundary node. Because boundary nodes do not fire, there can be no chips passing from one boundary node to another. Thus it is reasonable to have an entry of zero for a_{ij} when $s_i \in B$ and $i \neq j$. It may be desirable, however, to distinguish such entries from the diagonal entries a_{ii} for boundary nodes s_i, because boundary nodes, while they do not fire, do retain all of their chips. Since the number of chips on a boundary node may increase as the game progresses, there is no single number we can use as the diagonal entry for boundary nodes. This unfortunate fact means that trying to use the number of chips that fire between nodes as the matrix entry will not allow us to form one matrix that completely describes chip movement in the game. There is another more subtle problem with this approach. Suppose for the moment that we were to double the number of arrows between every pair of nodes. Then it would follow that an interior node s_i would fire when there were four chips on it, and two chips would pass to each node receiving chips from s_i. Thus the matrix for the chip firing game would be

$$A = \left(\begin{array}{ccccc|cc} 0 & 2 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 2 \\ - & - & - & - & - & - & - \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right) \tag{2}$$

On the surface, it doesn't seem as though this would be a problem. However, we noted that the probabilities and expected value encountered in the example above were computed as proportions or ratios. The probability 1/4 of sequence HHT winning the game was given by the ratio of chips in node HHT to the number of chips added to the START node. The probability 3/4 was similarly computed. The expected value of 6.5 tosses was calculated by taking the total number of moves and dividing by the total number of chips added to the START node, four. Suppose we were to play the chip firing game with our new graph (having doubled the number of arrows between each pair of nodes) starting with the critically loaded configuration, that is three chips on each interior node. Here is what we would find: adding eight chips to the START node

would result in two chips ending up on node HHT, six chips on node THH, and a total of 52 moves (check this for yourself by playing the game). This yields ratios of $2/8 = 1/4$ and $6/8 = 3/4$ for nodes HHT and THH respectively. Also, the total number of moves divided by the number of chips added to the START node is $52/8 = 6.5$. [Do you think that you will end up with the same ratios if you triple the number of arrows between each pair of nodes in the original graph? What if you multiply the number of arrows between each pair by n? These questions will be explored in the exercises at the end of this section.] It seems as though what is important in these calculations is the proportion of chips that pass from one node to another during a firing, rather than the quantity of chips. We will redefine our matrix by requiring that the ij entry contain the proportion of the chips from node s_i that pass to node s_j when s_i fires. Let that proportion be denoted by p_{ij}. We can call the p_{ij} *transition proportions*, because they describe the transition from one configuration to another when firings occur. This new approach also provides a sensible solution for the problem of how to define the entries for boundary nodes. Because the boundary nodes never fire, we can now define p_{ij} for $s_i \in B$ as follows: If $s_i \in B$ and $i \neq j$, then s_j will never receive any chips from s_i. Therefore it makes sense to define $p_{ij} = 0$ in this case (as we did before for matrix A). However, in order to highlight the fact that the boundary nodes retain all of their chips, we can define $p_{ii} = 1$ for $s_i \in B$. So we no longer have to worry about the fact that the number of chips on a boundary node may increase as the game progresses. Each boundary node will retain 100% of its chips. For our example of the coin tossing game between HHT and THH, the set of nodes in the finite graph are $s_1 =$ START, $s_2 =$ H, $s_3 =$ T, $s_4 =$HH, $s_5 =$TH, $s_6 =$ HHT, $s_7 =$ THH. The boundary nodes are s_6 and s_7, precisely the sequences at which the game ends. The other five nodes are interior. Since we played this game with a fair coin all transition probabilities are either 0 or $\frac{1}{2}$. The transition matrix is

$$P = \left(\begin{array}{ccccc|cc} 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 \\ - & - & - & - & - & - & - \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right) \tag{3}$$

Using transition probabilities for any specific game, we now have a single matrix whose entries give a complete description of how chips are fired from node to node in the game.

5.2 Analysis of Coin Tossing Games as Markov Processes

In the previous section we saw how a matrix of proportions can render a faithful picture of the movement of chips in a chip firing model of coin tossing. We are now ready to complete the mathematical description and analysis of chip firing games as mathematical structures that have come to be known as Markov processes, named to honor the Russian mathematician A. A. Markov who first introduced the central ideas of these models in the first decade of the twentieth century. The first step in analyzing our model is to give a clear description of how coin tossing games and their chip firing models can be defined as types of Markov processes. Once we have accomplished this task we will use standard Markov techniques to prove that chip firing models give exact answers to coin tossing type games. Having done this in sufficient generality we will extend the discussion to more elaborate games that can be thought of as modified coin tossing games. In all coin tossing games we have a finite set of *states*, $s_1, s_2, \ldots, s_n$, together with a set of *transition probabilities*

defined for pairs of states. The states correspond to the nodes in the finite graph of the coin tossing game. The transition probabilities are p and $1-p$ where p is the probability of tossing a head on any given flip. For $1 \leq i, j \leq n$, the transition probability p_{ij} has one of three values depending on the relationship between the sequences of Hs and Ts represented by s_i and s_j. To make the relationship easy to write down we will use S_i to name the sequence of Hs and Ts corresponding to state s_i. The symbols S_iH and S_iT then become the two possible sequences resulting from the next coin toss. Now we can define the transition probabilities for any coin tossing game.

Definition 1. In a coin tossing game with Prob(H) $= p$ and with states $s_1, s_2, \ldots, s_n$, for $1 \leq i, j \leq n$, the transition probability p_{ij} is defined by

$$p_{ij} = \begin{cases} p, & \text{if } S_j \text{ is } S_iH; \\ 1-p, & \text{if } S_j \text{ is } S_iT; \\ 0. & \text{otherwise.} \end{cases} \tag{4}$$

The $n \times n$-matrix of transition probabilities is denoted by $P = [p_{ij}]$ and is called the *transition matrix*.

Definition 2. In a coin tossing game with states $s_1, s_2, \ldots, s_n$,

1. a state s_i is called *interior* if there is a state s_j for which $p_{ij} > 0$. The set of all interior states is denoted I.

2. a state s_i is called *boundary* if for all states s_j, $p_{ij} = 0$. The set of all boundary states is denoted B.

Because $p_{ii} = 1$ for any $s_i \in B$, and because $p_{ij} = 0$ whenever $s_i \in B$ and $s_j \in I$, it is convenient to organize the transition matrix P by listing all interior states first, followed by all boundary states: $I = \{s_1, s_2, \ldots, s_r\}$ and $B = \{s_{r+1}, s_{r+2}, \ldots, s_n\}$. Using this ordering of states, P has the following block structure

$$P = \left(\begin{array}{c|c} Q & R \\ - & - \\ 0 & I \end{array}\right) \tag{5}$$

where 0 is an $(n-r) \times r$ matrix of all zeroes, while I is the $(n-r) \times (n-r)$ identity matrix. Note that the $(r \times r)$-matrix Q gives the same transition probabilities as P, but only between interior states. All these new definitions are illustrated in the next example.

Example 1. For our previous example of the coin tossing game between HHT and THH, the states (exactly the nodes in the finite graph) are $s_1 =$ START, $s_2 =$ H, $s_3 =$ T, $s_4 =$HH, $s_5 =$TH, $s_6 =$ HHT, $s_7 =$ THH. The boundary states are s_6 and s_7, precisely the sequences at which the game ends. The other five states are interior. Since we played this game with a fair coin all transition probabilities are either 0 or $\frac{1}{2}$. The transition matrix is

$$P = \left(\begin{array}{ccccc|cc} 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 \\ - & - & - & - & - & - & - \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right) \tag{6}$$

In our analysis of coin tossing games the entries of matrices P^n and Q^n will play a pivotal role. The entry in the i^{th} row and j^{th} column of P^n is denoted $p_{ij}^{(n)}$. Symbolically, $P^n = [p_{ij}^{(n)}]$. Note carefully the use of (n) as a superscript—not an exponent. The ij entry of P^n (here n is an exponent) is not p_{ij} raised to the n^{th} power, and so we can't use p_{ij}^n as the symbol for the ij element of P^n.

Theorem 3. *For transition matrix P, $p_{ij}^{(n)}$ gives the probability that the coin tossing game is in state s_j exactly n tosses after having been in state s_j.*

Proof. When we write out the formula for the ij entry of P^n according to a recursive definition of matrix multiplication, we see that $p_{ij}^{(n)} = \sum_k p_{ik}^{(n-1)} p_{kj}$. If $k \notin I$, then by definition of I, $p_{kj} = 0$ for all values of j. So for all $k \notin I$, $p_{ik}^{(n-1)} p_{kj} = 0$. Thus for any term of P^n, we can restrict the summation to interior states without changing the term's value. In other words, $p_{ij}^{(n)} = \sum_{k \in I} p_{ik}^{(n-1)} p_{kj}$.

Incidentally, there's a related and well-known graph theory result that the ij entry in n^{th}-power of the incidence matrix gives the number of $i-j$ paths in the graph. See [14] for details. □

Because Q is an upper-left square submatrix of P we know that the ij^{th} element of Q is the ij^{th} element of P. Therefore the elements of Q give the same information as those in P. The only difference is that in Q we have information about interior states only. As it turns out we need to analyze the long-term behavior of only these states in order to determine the long-term probabilities for the boundary states. To complete the calculation of long-term absorption probabilities we will use $f_{ij}^{(n)}$ to denote the probability that the game is in boundary state s_j **exactly** n tosses after having been in interior state s_i. By the same argument that we used in the previous proof we know that $f_{ij}^{(n)} = \sum_{k \in I} p_{ik}^{(n-1)} p_{kj}$. In this sum, we again can restrict our attention to $k \in I$ because every sequence of n tosses that begins in interior state s_i and ends in boundary state s_j on exactly the n^{th} toss must be in some interior state s_k on the $(n-1)^{st}$ toss.

We can put all this information in a single matrix $F^{(n)} = [f_{ij}^{(n)}]$ where $i \in I$ and $j \in B$. Again note that $F^{(n)}$ is not the n^{th} power of F. The product $F \times F$ isn't even defined because F is not square. In our example $F^{(1)} = R$. The ij-entry of $F^{(1)}$, $f_{ij}^{(1)}$, is the probability that a sequence of tosses moves from interior state s_i to boundary state s_j in exactly one toss. Continuing with our game, $F^{(2)} = QR$, and the ij-entry of $F^{(2)}$, $f_{ij}^{(2)}$, is the probability that a sequence of tosses moves from interior state s_i to boundary state s_j is exactly two tosses. Specifically,

$$F^2 = QR = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1/2 & 0 \\ 0 & 1/2 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 1/4 & 0 \\ 0 & 1/4 \\ 1/4 & 0 \\ 0 & 0 \end{pmatrix} \tag{7}$$

Note that $f_{46}^{(2)} = 1/4 \neq 3/4 = p_{46}^{(2)}$. To see why this is so first note that $f_{46}^{(2)}$ is the probability of our coin tossing game moving from s_4 to s_6 **for the first time** in exactly two tosses. In other words, the game must move from s_4 to some interior state and then on to s_6. In contrast $p_{46}^{(2)}$ is the probability of the game moving from s_4 to s_6 in exactly two tosses including the possibility that the game was already in boundary state s_6 before the second toss. The reason that $F^{(n)}$ probabilities are important is because for a fixed ij entry, the events whose probabilities are $f_{ij}^{(1)}$, $f_{ij}^{(2)}$, …, $f_{ij}^{(k)}$ are mutually exclusive events. Specifically, each term $f_{ij}^{(1)}$, $f_{ij}^{(2)}$, …, $f_{ij}^{(k)}$, gives the probability that starting from interior state s_i the game enters boundary state s_j for the first time in one move, two

moves, ..., k moves, respectively. Since these are mutually exclusive events, we simply add $f_{ij}^{(k)}$, $1 \leq k$, to get the probability that starting from interior state s_i the game ends in boundary state s_j in some finite number of tosses.

So we see that we have only to sum $F^{(n)}$, for all $n \geq 1$, to get the long term absorption probabilities. Notice that this sum is also an $r \times (n-r)$ matrix. We will refer to elements of this sum matrix as f_{ij}, using no superscript. Specifically, the ij^{th} element of $\sum_{n=1}^{\infty} F^{(n)}$, denoted f_{ij}, gives the probability that the coin tossing game ends boundary state s_j after having started in interior state s_i. Notice that this infinite series can be rewritten (because R is constant in the series) as a series in powers of Q:

$$\sum_{n=1}^{\infty} F^{(n)} = \sum_{n=1}^{\infty} Q^{n-1}R = \Big(\sum_{n=1}^{\infty} Q^{n-1}\Big)R \tag{8}$$

So the problem of determining the long-term absorption probabilities for the coin tossing game has be reformulated in terms of the infinite matrix series $\sum_{n=1}^{\infty} Q^{n-1}$. Because all our calculations are based on this infinite sum, the first task is to verify that the series actually converges. The following lemma gives a sufficient condition for convergence.

Lemma 4. *Let A be a square matrix. If A^n tends to the zero matrix ($\forall i, j$, $a_{ij}^{(n)} \to 0$ as $n \to \infty$), then matrix $(I - A)$ has an inverse and*

$$(I - A)^{-1} = I + A + A^2 + \cdots = \sum_{k=0}^{\infty} A^k.$$

Proof. Consider the identity

$$(I - A)(I + A + A^2 + \cdots + A^{n-1}) = I - A^n.$$

For sufficiently large n, $I - A^n$ is arbitrarily close (entry-by-entry) to I and so has non-zero determinant. Hence neither factor of the product on the left side can have zero determinant. So $I - A$ has non-zero determinant and $(I - A)^{-1}$ exists. Multiply both sides of equation by this inverse and note again that $(I - A^n)$ tends to I. □

The next lemma shows that Q^n satisfies the hypotheses of the previous lemma. Thus we will be able to conclude that the series $\sum_{n=1}^{\infty} Q^{n-1}$ does, in fact, converge.

Lemma 5. *For the transition matrix P of a coin tossing game,*

$$p_{ij}^{(n)} \to 0, \forall s_i, s_j \in I$$

as $n \to \infty$.

Proof. Let $s_i \in I$. Starting in s_i it must be possible to reach a boundary state in not more than m tosses. So there is a number t, $0 < t < 1$, such that the probability of reaching a boundary state in no more than m tosses is at least t. Hence the probability of *not* reaching any boundary state in m tosses is at most $1 - t$, $0 < (1 - t) < 1$. Then the probability of not reaching a boundary state (i.e., remaining in interior states) from state s_i in nm independent tosses is no more than $(1 - t)^n$. Since $(1 - t)$ is between 0 and 1, this probability tends to 0 as $n \to \infty$. □

Now we can combine the two lemmas to get this crucial information about Q^n.

Theorem 6. *For any coin tossing game, $I - Q$ has an inverse and*

$$(I - Q)^{-1} = I + Q + Q^2 + \cdots = \sum_{k=0}^{\infty} Q^k.$$

Not only do we now know that $\sum_{k=0}^{\infty} Q^k$ exists, we have an effective method of calculating it. Remember the equation that gives long-term absorption probabilities is

$$\sum_{n=1}^{\infty} F^{(n)} = \Big(\sum_{n=1}^{\infty} Q^{n-1}\Big) R.$$

By simple substitution we now know that the probability a coin tossing game ends in boundary state s_j having started in interior state s_i is given by

$$\sum_{n=1}^{\infty} F^{(n)} = (I - Q)^{-1} R.$$

Let's see how all this calculation works out in the coin tossing we have been studying: HHT vs. THH.

Example 2. The matrix Q is the upper-left 5×5 submatrix of P in (6). Subtracting Q from the 5×5 identity matrix, we get

$$I - Q = \begin{pmatrix} 1 & -1/2 & -1/2 & 0 & 0 \\ 0 & 1 & -1/2 & -1/2 & 0 \\ 0 & 0 & 1/2 & 0 & -1/2 \\ 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & -1/2 & 0 & 1 \end{pmatrix} \tag{9}$$

Using a calculator, it is easy to find

$$(I - Q)^{-1} = \begin{pmatrix} 1 & 1/2 & 3 & 1/2 & 3/2 \\ 0 & 1 & 2 & 1 & 1 \\ 0 & 0 & 4 & 0 & 2 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 & 2 \end{pmatrix} \tag{10}$$

Now it's straightforward to calculate

$$F = \sum_{n=1}^{\infty} F^{(n)} \tag{11}$$

$$= (I - Q)^{-1} R \tag{12}$$

$$= \begin{pmatrix} 1 & 1/2 & 3 & 1/2 & 3/2 \\ 0 & 1 & 2 & 1 & 1 \\ 0 & 0 & 4 & 0 & 2 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 & 2 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1/2 & 0 \\ 0 & 1/2 \end{pmatrix} = \begin{pmatrix} 1/4 & 3/4 \\ 1/2 & 1/2 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{13}$$

The entries f_{11} and f_{12} in the product matrix give the familiar probabilities of winning for HHT and TTH, respectively. In terms of boundary states, these values are the long-term probabilities of being in boundary states s_6 and s_7, respectively, having started in interior state s_1. In general, the ij entry in F gives the long-term probability of being in boundary state s_{r+j} (Do you see why the r is added?) having started in interior state s_i.

In addition to making it easy to calculate the probabilities of absorption, the matrix $(I-Q)^{-1}$ gives other important information about the duration of the coin tossing game. This matrix is important enough in its own right to warrant notation that's easier to write than $(I-Q)^{-1}$. We will use $T = (I-Q)^{-1}$. A key observation about T is that it satisfies the identity $T = I + QT$. It's easy to prove this by noting that since $T = I + Q + Q^2 + \cdots$, we also know $QT = Q + Q^2 + +Q^3 \cdots$. Subtracting these two expressions yields I, and rewriting $T = I + QT$ as $T - QT = I$ gives the required identity. Let's first look at the off-diagonal entries of T. For all $i \neq j$, $1 \leq i, j \leq r$ the ij element of T is given by

$$t_{ij} = \sum_{k \in I} p_{ik} t_{kj}.$$

Written in this form t_{ij} looks like a weighted sum in which the weights are probabilities, i.e. an expected value for some random variable. In fact, t_{ij} can be thought of as the expected number of tosses that put the game in interior state s_j provided the game has started in interior state s_i. In order to see how this interpretation makes sense, we first note that for our matrices T and Q, we know that $TQ = QT$. Although matrix multiplication is not commutative in general, in this case since T is just a sum of *powers* of Q, it's easy to prove that $TQ = QT$. Thus $I + QT = I + TQ$, and so t_{ij} can also be written as

$$t_{ij} = \sum_{k \in I} t_{ik} p_{kj}.$$

Using this expression for t_{ij}, we will be able to see how t_{ij} can be interpreted as an expected number of coin tosses. In our HHT vs. THH example, let's look first at t_{ij} values with $i = 1$. This just means the game starts in s_1, the natural starting place for a coin tossing game. Specifically, what does $t_{12} = 1/2$ mean, and how does the calculation $t_{12} = t_{11}p_{12}$ give us the expected number of tosses that result in the game being in s_2? First we know that $t_{11} = 1$ since for any game that starts in s_1, the first visit to s_1 is the only one. The statement that $t_{12} = (1/2)t_{11}$ simply reflects the fact that whenever s_1 is visited, one half of the next tosses result in the game moving to s_2. Substituting $t_{11} = 1$, we get $t_{12} = 1/2$ which can now be interpreted as indicating that half the games we play starting at s_1 pass through s_2. An alternative way of interpreting the same information is simply to say that the expected number of visits to s_2 in any completed game is $1/2$. In these interpretations we are using the fact (which is not true for all states in our game) that any completed game visits s_2 exactly once. To see an example where a state can be visited many times in a single game, let's look at s_4 and how the calculation for t_{14} works. When the game starts in s_1 there are two ways to get to s_4. The game can visit s_4 from s_2 or from s_4. In terms of coin tosses, entering from s_2 corresponds to tossing the second consecutive H, while entering from s_4 corresponds to each subsequent toss of an H. Reflecting these two ways of visiting s_4, the expected number of times the game is in s_4 is a sum: $1/2$ the number of times it is in s_2 plus $1/2$ the number of times it is in s_4. Hence $t_{14} = (1/2)t_{12} + (1/2)t_{14} = 1/4 + (1/2)t_{14}$, which gives $t_{14} = 1/2$. Note that $t_{14} = 1/2$ does not mean that $1/2$ the games visit s_4. Only half the games that visit s_2 go on to s_4, the others move to s_3 and cannot subsequently get to s_4. So the proportion of games that visit s_4 is certainly less than $1/2$. Because a single game can visit s_4 repeatedly in the course of play, we must be careful this time to refer to t_{14} as the expected number of tosses during a completed game in which the game is in s_4, in other words, the expected number of visits to state s_4.

Activity 5.2

Students should work in pairs or triples. Each group will toss a coin and record the result of each toss. When either of the two sequences HHT or THH appears as the

last three tosses, stop the game. (So far this is the same as an earlier activity.) This time add the following directions for record-keeping. As each toss is made keep track of the interior state of the game (s_1, s_2, ..., s_5). At the end of each game record the number of times the game was in each interior state. Repeat this process a total of five times for each group. Now the instructor will collect information from each group and pool the results. The average number of visits to each state can now be calculated and compared to the first row of the matrix $(I - Q)^{-1}$. Students can also collect and report information about the NUMBER of games that visited s_4, for example, and using pooled data calculate the proportion of games that visited s_4.

Now we're ready to tackle t_{13} and t_{15}. Because the game can cycle between s_3 and s_5, we'll do these two together. The only way a game that starts in s_1 can get to s_5 is through s_3, and exactly half the time the game is in s_3 it proceeds directly to s_5. Hence

$$t_{15} = (1/2)t_{13}. \tag{14}$$

Next notice that s_3 can be entered from a number of different states. For a game that starts in s_1, the game can progress to s_3 directly from s_1, from s_2, from itself, and from s_5. These approaches are mutually exclusive and so the expected number of visits to s_3 should be

$$t_{13} = (1/2)t_{11} + (1/2)t_{12} + (1/2)t_{13} + (1/2)t_{15} = (1/2)(1) + (1/2)(1/2) + (1/2)t_{13} + (1/2)t_{15} \tag{15}$$

Solving the system of equations given by (14) and (15), we get $t_{13} = 3$ and $t_{15} = 3/2$. So now we know what each of the elements in the first row of matrix $(I - Q)^{-1}$ represent. But what about the other rows? How can we think of them? Well , it's simply a matter of assuming you start the coin tossing game with a given non-winning string of Hs and Ts. For example, consider t_{24}. To understand this entry, we assume the coin tossing game starts with sequence H already written. So $t_{22} = 1$ because if a game starts in s_2, that's the only time it will be in s_2: there's no way to revisit. As when we started in s_1, there are two ways to get to s_4. So t_{24} is 1/2(number of visits to s_2) plus 1/2(number of visits to s_4). Using $t_{22} = 1$, we get $t_{24} = 1/2(1) + (1/2)t_{24}$ which gives $t_{24} = 1$. The calculations and interpretations for t_{23} and t_{25} are similar to those for t_{13} and t_{15}, but you must remember that the game starts in s_2. So there are only three ways to visit s_3. In fact, since a game beginning in s_2 never visits s_1, we know $t_{21} = 0$. You should try to complete the calculations for t_{23} and t_{25}.

Exercises

1. Consider the graph for the coin tossing game between sequences HHT and THH. Suppose that we triple the number of arrows between each pair of nodes. Determine (by playing the game) whether we end up with 1/4 and 3/4 as the ratios of the numbers of chips on nodes HHT and THH to the number added to the START node, respectively. What do we get if we divide the total number of moves by the number of chips added to the START node? Be sure to start the game in a critically loaded configuration and play until the critically loaded configuration is reached again.

2. (Refer to the previous exercise) Will the ratios remain the same if we multiply the number of arrows between each pair of nodes by the positive integer n rather than by 2 or 3? Explain your reasoning.

3. For the coin tossing game HHH vs. HTH, draw the finite graph of the game (use s_1 as a start node), form matrices P, Q, and R, and use a calculator or some software to find

$T = (I - Q)^{-1}$. Use all this information to determine the probability that HHH wins. You should be able to show that HHH has less than 1/2 chance of winning this game. Will starting the game assuming an H has already been tossed improve the chances that HHH wins? If so, by how much?

4. Draw the probability tree for six tosses in the game HHT vs. THH. Label the nodes that correspond to s_3 and s_5. Use this tree to argue that $t_{15} = 1/2t_{13}$.

5.3 Using Markov Analysis in More Elaborate Games

Some well known children's games which at first glance seem much more elaborate than coin tossing can, in fact, be analyzed using standard Markov techniques (see [1, 6, 9, 11, 12]). Remember *Chutes and Ladders*®? Each player rolls a single die and moves his or her game piece the indicated number of steps along a board with 100 squares numbered in order. The first player whose piece reaches square number 100 exactly wins. Since each player's position on the board is completely determined by his or her dice rolls this game can be analyzed as a Markov chain. All transition probabilities are 0 or $\frac{1}{6}$. Of course, there are the chutes and ladders of the game's title: 10 chutes and 9 ladders. If at the end of your move for a particular roll of the die, your game piece lands on the square at the bottom of a ladder, your piece climbs the ladder and actually ends your turn on the square at the top of the ladder. If your game piece lands on the square at the top of a chute, your piece slides down the chute and end up on the square at the bottom of the chute. If you think about this for a minute you will realize that there are exactly $100 + 1 - (10 + 9)$ states in this game; all but two (START and 100) are interior states. The matrix of transition probabilities for *Chutes and Ladders*® would be an 82×82 matrix. Determining $(I - Q)^{-1}$ in order to compute the expected length of a game would involve inverting a very large, although very sparse, matrix. A smaller, but more complicated game for which we can complete the calculations is Parker Brothers' *Hi Ho! Cherry-O*®. The object of the game is to pick all ten cherries from your tree and fill your bucket. Each player has his own tree and his own bucket. On a turn in this game, the player spins a spinner that has seven equally likely regions. If the spinner indicates one, two, three, or four cherries, the player moves that many cherries from his tree to his bucket. If the spinner lands on the section with a bird or on the section with a dog, then the player takes two cherries from his bucket, and puts them back on his tree. (If, in the case of a dog or bird spin, the player has one cherry, he puts it back; if he has no cherries, he does nothing.) If the spinner lands on the spilled bucket section, then the player puts ALL the cherries in his bucket back on his tree. The states, of course, are the numbers $0, 1, 2, \ldots, 10$ corresponding to the number of cherries in the player's bucket. If the spinner lands on a number that is greater than the number of cherries remaining on the player's tree, then the player moves all of his remaining cherries to his bucket and wins the game.

We can also use the matrix T to determine the expected length of any game provided we know the state in which the game starts. To see how this works, think about a game that starts in state s_i. Then $\sum_{k \in I} t_{ik}$ is just the expected number of times each interior state is visited in any completed game that starts in state s_i. Thus the sum of all entries in row i of T gives precisely the total number of times a completed game visits any interior state. In other words, the sum of the entries in row i gives the expected number of plays of the game before absorption.

Activity 5.3

1. Construct the transition matrix for *Hi Ho! Cherry-O*®.
   (a) Determine the expected number of spins for one player to fill his bucket assuming he starts with an empty bucket.

(b) Determine the expected number of spins for one player to fill his bucket assuming he starts with four cherries.

2. Johnson [11] has also analyzed a game called *Drop Dead*. Find the rules for this game, construct the matrix of transition probabilities, and determine the expected number of rolls on a player's turn.

3. (NOT for the faint of heart) Determine the expected number of turns for a single player to reach square number 100 in *Chutes and Ladders*®.

5.4 A Glimpse into More Standard Applications of Markov Analysis as Chip Firing Games

Markov techniques are used in such fields as economics, natural sciences (including biology and ecology), social science, and business management to model a variety of systems that change according to some reasonably well-defined transition probabilities. To give an flavor of these applications we conclude with an example from biology. This treatment in this section follows both [14] and [13]. More complete and wide-ranging expositions of the ideas of this example and of other applications of Markov analysis can be found in both books.

We consider the inheritance of a single genetic trait that is governed by the simplest model of inheritance. In this model the inheritance of a single trait is determined by a pair of genes, one gene inherited from each parent. There are two forms of this gene: dominant and recessive. There are only two possible forms of the trait: something like two possible coat colors in a dog breed. We will denote the dominant form of the gene by A and the recessive form by b. This is not standard notation, but it will work well for our purposes. Although the animal has two genes for this trait, the animal will have the recessive trait only if it has inherited a recessive gene from each parent. For example, if one parent is AA (has two dominant genes for the trait—*pure dominant*) and the other if Ab (has one dominant and one recessive gene—*hybrid*) both parents will show as type A. The four equally likely offspring of a mating of these two animals will be AA, AA, Ab, and Ab. The states will be possible pairs of parents: (Ab, Ab), (AA, bb), (AA, Ab), (Ab, bb), (AA, AA), (bb, bb). Our example is based on the assumption that after any mating, the next generation of animals select mates according to whether the potential mate looks the same with respect to this one trait. Kemeny and Snell in [13] study a general problem in which an offspring is k times as likely to favor an animal that is unlike it. Hence k measures the degree to which "opposites attract". In our simpler model, however, an offspring is just as likely to pick an animal that looks different from it as it is to pick one that looks like it. For example, the generation of offspring from the parent-pair (AA, Ab) we looked at earlier is easy to analyze since all the offspring appear the same. Hence the possible parent-pairs are (AA, AA), (AA, Ab), and (Ab, Ab) with probabilities 1/4, 1/2, and 1/4, respectively. So the model we're looking at moves from state (AA, Ab) only to states (AA, AA), (AA, Ab), and (Ab, Ab) with these probabilities. Note that (AA, AA) and (bb, bb) are boundary states since all offspring of (bb, bb), for example, carry the gene pair bb. Thus the only possible parent-pairs of the next generation are also (bb, bb), and so all offspring will be bb-type animals.

If we list the states (pairs of parents) in the order given above then the matrix of transition

probabilities is

$$P = \begin{pmatrix} 1/4 & 1/8 & 1/4 & 1/4 & 1/16 & 1/16 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1/4 & 0 & 1/2 & 0 & 1/4 & 0 \\ 1/4 & 0 & 0 & 1/2 & 0 & 1/4 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{16}$$

We know that eventually we will find ourselves in the boundary states having developed pure strains for this trait. Two interesting questions suggest themselves:

1. What is the eventual proportion of the two pure strains and how does it depend on the starting state?
2. How long does it take (how many generations) to reach these proportions?

These questions can be answered analytically using all the algebra from the preceding Markov process section or we can think of these as chip firing games in which the pure strains become the winning or losing nodes of the chip firing graph. The following activity can be completed using either an algebraic approach or chip firing techniques.

Activity 5.4

All parts of this activity use the transition probabilities from the matrix P in equation 16 above.

1. Draw the chip firing diagram that corresponds to the matrix of transition probabilities.
2. Starting from parent-pair state (Ab, Ab) verify that the eventual proportions of pure dominant strains (AA, AA) and (bb,bb) will be 50-50. Does any feature of your chip firing diagram make this seem reasonable?
3. Starting from parent-pair state (Ab, AA) verify that the eventual proportions of pure dominant strains (AA, AA) and (bb,bb) will be 3/4 to 1/4, respectively. If you use chip firing (instead of algebraic) methods you should pre-load your diagram to form the critical configuration. Otherwise, it will take much too long to find a configuration that repeats.
4. Describe how you would determine the expected number of generations to reach a pure strain—parent-pairs (AA, AA) or (bb, bb)—if you start with (Ab, AA) or (Ab, bb).
5. Based on the chip firing diagram can you explain why the following statement is true? The expected number of generations needed to reach a pure strain from (AA, bb) is exactly one more than the number of generations needed to reach a pure strain from (Ab, Ab).

A Solutions to Selected Exercises

Section 1 Solutions

Exercise 1, Page 5 - Answers to this exercise will depend on the results each group or student obtained in Activity 1.1

Exercise 2, Page 5 - Answers will vary from student to student, but it appears as though the probability of winning each game is between 0.1 and 0.2.

Exercise 3, Page 5 - Answers will vary from student to student but given nothing more than the information in Tables 1 and 2, it seems reasonable to think that Lotto Game One offers a better chance of winning.

Exercise 4, Page 8 - The game can only be lost if the traffic light turns red (R) when the counter is on the \$1 node or the \$2 node. If the first or second flash is R, then we will lose. Otherwise, the first two flashes will be green (G), and we will be on the \$4 node. At this point, if either of the next two flashes (the third or fourth) is G then we will win because the counter will move to the \$5 node. Thus we assume that the third and fourth flashes are R, which moves us back to the \$1 node where we started. Thus, every losing path will result from a sequence of flashes of the form $(\text{GGRR})^i\text{R}$ or $(\text{GGRR})^i\text{GR}$ where i can be any nonnegative integer. Since the probability of R or G on any given flash is 1/2, the probability of a sequence of flashes of the form $(\text{GGRR})^i\text{R}$ is $(\frac{1}{2})^{4n+1}$ and the probability of a sequence of the form $(\text{GGRR})^i\text{GR}$ is $(\frac{1}{2})^{4n+1}$. Thus, since any nonnegative integer i is possible, and since either of the two sequence forms is possible, the probability of losing the game is given by

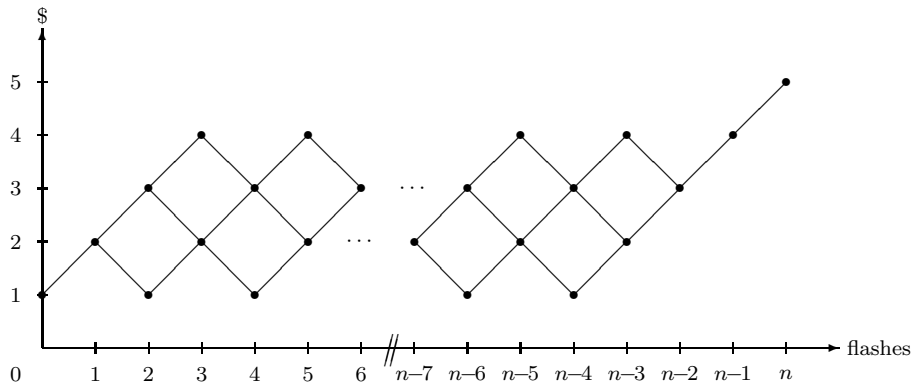
$$\begin{aligned}
P(\text{lose game}) &= \sum_{i=0}^{\infty}\left(\frac{1}{2}\right)^{4n+1} + \sum_{i=0}^{\infty}\left(\frac{1}{2}\right)^{4n+2} \\
&= \sum_{i=0}^{\infty}\left(\frac{1}{2}\right)\left(\frac{1}{2}\right)^{4n} + \sum_{i=0}^{\infty}\left(\frac{1}{4}\right)\left(\frac{1}{2}\right)^{4n} \\
&= \sum_{i=0}^{\infty}\left(\frac{1}{2}+\frac{1}{4}\right)\left(\frac{1}{16}\right)^{n} \\
&= \sum_{i=0}^{\infty}\left(\frac{3}{4}\right)\left(\frac{1}{16}\right)^{n} \\
&= \frac{\frac{3}{4}}{1-\frac{1}{16}} \\
&= \frac{3}{4}\cdot\frac{16}{15} \\
&= \frac{4}{5}
\end{aligned}$$

Exercise 5, Page 8 - The game can only be won if the traffic light turns green (G) when the counter is on the \$3 node or the \$4 node. If the first or second flash is red (R), then we will lose. Thus, in order to win, the first two flashes must be green (G), putting the counter on the \$4 node. At this point, if the third flash is G we win and if the third flash is R we move to the \$3 node. If we are on the \$3 node, then a G on the fourth flash will win the game and an

R on the fourth flash returns us to the \$1 node where we started. Thus, every winning path will result from a sequence of flashes of the form $(\text{GGRR})^i\text{GGG}$ or $(\text{GGRR})^i\text{GGRG}$ where i can be any nonnegative integer. Since the probability of R on any given flash is 3/5 and the probability of G on any given flash is 2/5, the probability of a sequence of flashes of the form $(\text{GGRR})^i\text{GGG}$ is $(\frac{2}{5})^{2n+3}(\frac{3}{5})^{2n}$ and the probability of a sequence of the form $(\text{GGRR})^i\text{GGRG}$ is $(\frac{2}{5})^{2n+3}(\frac{3}{5})^{2n+1}$. Thus, since any nonnegative integer i is possible, and since either of the two sequence forms is possible, the probability of losing the game is given by

$$\begin{aligned}
P(\text{win game}) &= \sum_{i=0}^{\infty}\left(\frac{2}{5}\right)^{2n+3}\left(\frac{3}{5}\right)^{2n}+\sum_{i=0}^{\infty}\left(\frac{2}{5}\right)^{2n+3}\left(\frac{3}{5}\right)^{2n+1}\\
&= \sum_{i=0}^{\infty}\left(\frac{2}{5}\right)^{3}\left(\frac{2}{5}\right)^{2n}\left(\frac{3}{5}\right)^{2n}+\sum_{i=0}^{\infty}\left(\frac{2}{5}\right)^{3}\left(\frac{3}{5}\right)^{1}\left(\frac{2}{5}\right)^{2n}\left(\frac{3}{5}\right)^{2n}\\
&= \sum_{i=0}^{\infty}\left(\frac{2}{5}\right)^{3}\left(\frac{6}{25}\right)^{2n}+\sum_{i=0}^{\infty}\left(\frac{2}{5}\right)^{3}\left(\frac{3}{5}\right)^{1}\left(\frac{6}{25}\right)^{2n}\\
&= \sum_{i=0}^{\infty}\left[\left(\frac{2}{5}\right)^{3}\left(1+\frac{3}{5}\right)\right]\left(\frac{36}{625}\right)^{n}\\
&= \sum_{i=0}^{\infty}\left(\frac{8}{125}\right)\left(\frac{8}{5}\right)\left(\frac{36}{625}\right)^{n}\\
&= \sum_{i=0}^{\infty}\left(\frac{64}{625}\right)\left(\frac{36}{625}\right)^{n}\\
&= \frac{\frac{64}{625}}{1-\frac{36}{625}}\\
&= \frac{64}{625}\cdot\frac{625}{589}\\
&= \frac{64}{589}
\end{aligned}$$

Exercise 6, Page 8 - The shortest winning sequence in this game is GGGG, the sequence consisting of 4 green flashes. Note that any winning sequence must contain four green (G) flashes in addition to one green flash for every red (R) flash in the sequence. Thus, the total number of flashes in a winning sequence will be of the form $2k+4$, where $k \geq 0$ is the number of red flashes in the sequence. This tells us that the length of a winning sequence will be even and greater than or equal to 4. Further, since on any given flash G and R are equally likely, the probability of a particular winning sequence of length n occurring is $(\frac{1}{2})^n$. What remains is to enumerate the winning paths of length n for all even $n \geq 4$, to compute the probability of winning the game with a path of length n for all possible winning path lengths, and to sum those probabilities over all possible winning path lengths. Consider the diagram below. The points in the diagram represent the possible positions in the game after a given number of flashes. The lattice of line segments is to be read from left to right. A line segment of positive slope represents a G and a line segment of negative slope represents an R.



The game is lost if a path reaches the horizontal axis (so we will not consider such paths) and the game is won when a path reaches a height of 5. We wish to count the number of paths resulting in a win in exactly n flashes, where n is an even number greater than or equal to 4. All such paths start at the point $(0, 1)$, never drop down to the horizontal axis, and reach a height of 5 for the first time at the point $(n, 5)$. From the diagram we can see that all winning paths of length n must pass through the point $(n-2, 3)$, and that after reaching this point we must finish with GG. Thus, we need only count the paths from $(0, 1)$ to $(n-2, 3)$ in order to enumerate the winning paths of length exactly n.

Each point $(k, 3)$ in the diagram (k even) can be reached in one of two distinct ways: from the point $(k-1, 2)$ followed by G or from the point $(k-2, 3)$ followed by GR. Similarly, each point $(k, 2)$ in the diagram (k odd) can be reached in one of two distinct ways: from the point $(k-1, 3)$ followed by R or from the point $(k-2, 2)$ followed by RG. It is clear that the points $(1, 2)$ and $(2, 3)$ can each be reached in only one way.

Let $N(k)$ be the number of paths from $(0, 1)$ to $(k, 3)$ if $k \leq n-2$ is even and let $N(k)$ be the number of paths from $(0, 1)$ to $(k, 2)$ if $k \leq n-3$ is odd. From the previous paragraph we see that $N(1) = 1$, $N(2) = 1$, and $N(k) = N(k-1) + N(k-2)$. This is identical to the recursive formula for the Fibonacci sequence F_k: $F_1 = F_2 = 1, F_k = F_{k-1} + F_{k-2}$. In other words, the number of paths from $(0, 1)$ to $(n-2, 3)$ is equal to F_{n-2}. Thus, the probability of winning Lotto Game Two in exactly n flashes (n even, $n \geq 4$)is $(\frac{1}{2})^n F_{n-2}$. This means that the probability of winning Lotto Game Two is given by

$$\sum \left(\frac{1}{2}\right)^n F_{n-2},$$

where the sum is taken over all even n with $n \geq 4$. This sum can be rewritten as

$$\sum_{k=1}^{\infty} \left(\frac{1}{2}\right)^{2k+2} F_{2k}$$

noting that $n \geq 4$ and n even implies that $n = 2k + 2$ for $k \geq 1$.

As the reader may verify, there is an explicit formula for the n-th term of the Fibonacci sequence given by

$$F_n = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n \right].$$

If we let $\varphi = \frac{1+\sqrt{5}}{2}$, then $1-\varphi = \frac{1-\sqrt{5}}{2}$, and so the probability of winning Lotto Game Two is given by

$$\begin{aligned}
\sum_{k=1}^{\infty} \left(\frac{1}{2}\right)^{2k+2} \frac{1}{\sqrt{5}} \left[\varphi^{2k} - (1-\varphi)^{2k}\right] &= \sum_{k=1}^{\infty} \frac{1}{4\sqrt{5}} \left[\left(\frac{\varphi}{2}\right)^{2k} - \left(\frac{1-\varphi}{2}\right)^{2k}\right] \\
&= \sum_{k=1}^{\infty} \frac{1}{4\sqrt{5}} \left[\left(\frac{\varphi^2}{4}\right)^{k} - \left(\frac{(1-\varphi)^2}{4}\right)^{k}\right] \\
&= \sum_{k=1}^{\infty} \frac{\varphi^2}{16\sqrt{5}} \left(\frac{\varphi^2}{4}\right)^{k-1} - \sum_{k=1}^{\infty} \frac{(1-\varphi)^2}{16\sqrt{5}} \left(\frac{(1-\varphi)^2}{4}\right)^{k-1} \\
&= \frac{\varphi^2}{16\sqrt{5}} \left(\frac{1}{1-\frac{\varphi^2}{4}}\right) - \frac{(1-\varphi)^2}{16\sqrt{5}} \left(\frac{1}{1-\frac{(1-\varphi)^2}{4}}\right).
\end{aligned}$$

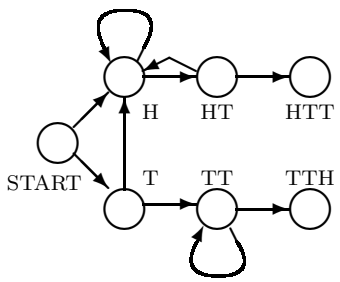
Recalling that $\varphi = \frac{1+\sqrt{5}}{2}$ and $1-\varphi = \frac{1-\sqrt{5}}{2}$, we will leave it to the reader to verify that $\varphi^2 = \frac{3+\sqrt{5}}{2}$, $(1-\varphi)^2 = \frac{3-\sqrt{5}}{2}$, $\frac{1}{1-\frac{\varphi^2}{4}} = \frac{8}{5-\sqrt{5}}$, and $\frac{1}{1-\frac{(1-\varphi)^2}{4}} = \frac{8}{5+\sqrt{5}}$.

Given this information, we have that the probability of winning Lotto Game Two is

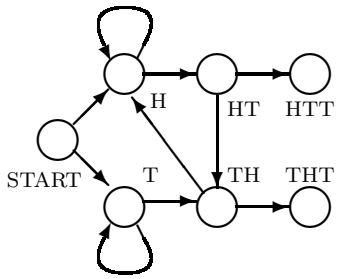
$$\begin{aligned}
\frac{\varphi^2}{16\sqrt{5}} \left(\frac{1}{1-\frac{\varphi^2}{4}}\right) - \frac{(1-\varphi)^2}{16\sqrt{5}} \left(\frac{1}{1-\frac{(1-\varphi)^2}{4}}\right) &= \frac{3+\sqrt{5}}{32\sqrt{5}} \left(\frac{8}{5-\sqrt{5}}\right) - \frac{3-\sqrt{5}}{32\sqrt{5}} \left(\frac{8}{5+\sqrt{5}}\right) \\
&= \frac{(3+\sqrt{5})(5+\sqrt{5}) - (3-\sqrt{5})(5-\sqrt{5})}{4\sqrt{5}(5-\sqrt{5})(5+\sqrt{5})} \\
&= \frac{(20+8\sqrt{5}) - (20-8\sqrt{5})}{80\sqrt{5}} \\
&= \frac{16\sqrt{5}}{80\sqrt{5}} \\
&= \frac{1}{5}.
\end{aligned}$$

Finally, we have that the probability of winning Lotto Game Two is $\frac{1}{5}$.

Exercise 7, Page 10 - Finite diagram for HTT vs.TTH:

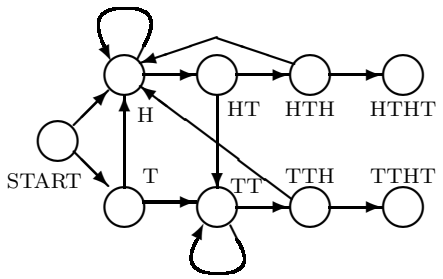


Exercise 8, Page 10 - Finite diagram for THT vs. HTT:



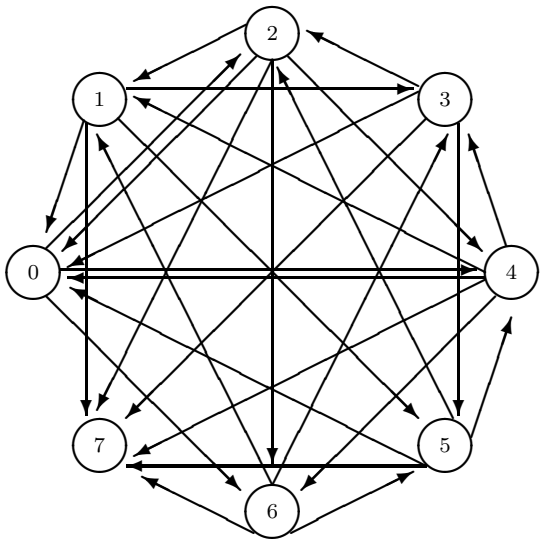
Exercise 9, Page 10 - Parts (a), (b), and (e) all give possible complete game sequences. In each case, either HHT or THH appears for the first time in the last three tosses. Sequence (c) is possible as a partial game sequence, but not a complete game sequence. Neither HHT nor THH has appeared when sequence (c) ends. Sequence (d) is not a possible game sequence because HHT appears after four tosses, thus ending the game, yet the sequence continues on after that.

Exercise 10, Page 10 - Finite diagram for HTHT vs.TTHT:



Exercise 11, Page 10 - If Ben had the sequence HHT and Jerry had the sequence HHTT, then it would be impossible for Jerry to win, for if the first three tosses in Jerry's sequence appear, then Ben wins automatically. Another problem could arise in which one player has sequence HTHT and the other has sequence THT. If the sequence HTHT comes up in this game, then it would appear that both players win simultaneously.

Exercise 12, Page 10 - Finite diagram for the dice game:

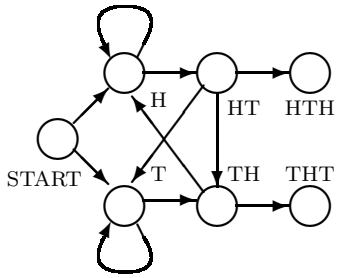


Exercise 13, Page 12 - There are only two possible ways to get from START to THT: START-T-...-T-TH-THT and START-H-T-...-T-TH-THT. In both cases, T-...-T represents looping at T any number of times. Thus, the probability of THT winning this game is given by

$$\begin{aligned}\left(\frac{1}{2}\right)^3\sum_{i=0}^{\infty}\left(\frac{1}{2}\right)^i+\left(\frac{1}{2}\right)^4\sum_{i=0}^{\infty}\left(\frac{1}{2}\right)^i &= \left(\left(\frac{1}{2}\right)^3+\left(\frac{1}{2}\right)^4\right)\sum_{i=0}^{\infty}\left(\frac{1}{2}\right)^i\\ &= \frac{3}{16}\sum_{i=0}^{\infty}\left(\frac{1}{2}\right)^i\\ &= \frac{3}{16}(2)\\ &= \frac{3}{8}\end{aligned}$$

as desired.

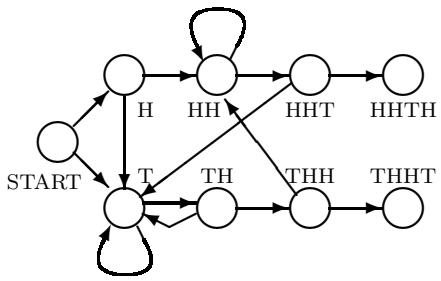
Exercise 14, Page 12 - Finite diagram for HTH vs.THT:



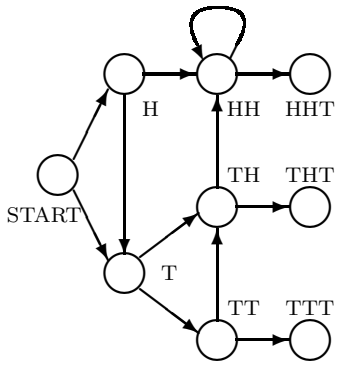
This game is difficult to analyze using the method of the example in the text because it is possible to bounce back and forth between the upper and lower halves of the diagram any number of times. This introduces a greater degree of complexity to the analysis than was present in the text example. However, we note that the diagram for this game is symmetric, leading us to believe that the probability of either sequence winning (HTH or THT) is $\frac{1}{2}$.

Exercise 15, Page 12 - An analysis of this game would present much the same difficulty as that of Exercise 14. As can be seen in the finite diagram for this game (given below), a winning path for either sequence can bounce back and forth between the upper and lower halves of the diagram any number of times. This will make enumerating winning paths for either sequence very challenging.

Finite diagram for HHTH vs. THHT:

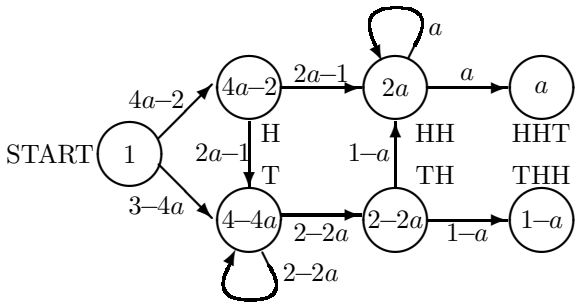


Exercise 16, Page 13 - Finite diagram for HHT vs. THT vs.TTT:



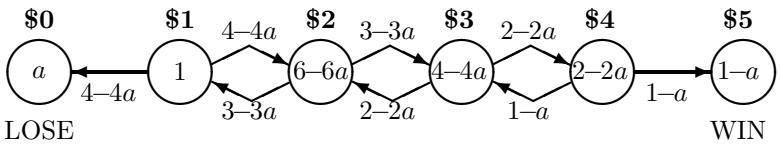
Section 2 Solutions

Exercise 1, Page 15 - If we let a represent the probability of sequence HHT appearing first, then $1 - a$ will be the probability of sequence THT appearing first. By backtracking we obtain the following diagram:



Since heads and tails are equally likely on each toss, we know that the probability of heads and the probability of tails on the first toss will both be equal to 1/2. Solving either of the two equations $4a - 2 = \frac{1}{2}$ or $3 - 4a = \frac{1}{2}$ yields the solution $a = 5/8$, as desired.

Exercise 2, Page 15 - If we let a represent the probability of losing the game (in other words, reaching the \$0 node), then $1 - a$ will be the probability of winning (reaching the \$5 node). By backtracking (beginning at the \$5 node) we obtain the following diagram:

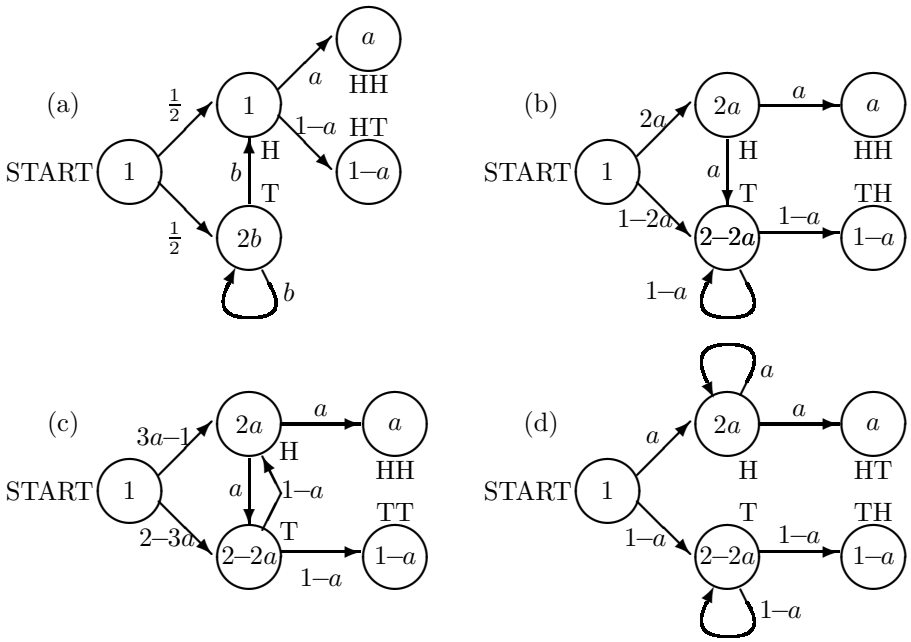


The \$0 node and the arrow entering it give us the equation $a = 4 - 4a$, which yields the solution $a = 4/5$.

Exercise 3, Page 15 - The table below gives the probability of the row sequence beating the column sequence. The calculations are provided below the table.

	HH	HT	TH	TT
HH	...	1/2	1/4	1/2
HT	1/2	...	1/2	3/4
TH	3/4	1/2	...	1/2
TT	1/2	1/4	1/2	...

- $P(\text{HH beats HT}) = 1/2$: If we let $a = P(\text{HH beats HT})$ and $1 - a = P(\text{HT beats HH})$, then backtracking gives us diagram (a) below. Since we don't know the value along the arrow from T to H, we label it b. Thus, the arrow from T to T is also labelled b and Node T gets a value of $2b$. Since the arrow from START to T has value 1/2, we get that $\frac{1}{2} + b = 2b$, or $b = \frac{1}{2}$. In any case, since Node H has value 1, we get that $a = P(\text{HH beats HT}) = \frac{1}{2}$.
- $P(\text{HH beats TH}) = 1/4$: If we let $a = P(\text{HH beats TH})$ and $1 - a = P(\text{TH beats HH})$, then backtracking gives us diagram (b) below. This gives us the equations $2a = \frac{1}{2}$ and $1 - 2a = \frac{1}{2}$, both of which yield the solution $a = \frac{1}{4}$.
- $P(\text{HH beats TT}) = 1/2$: If we let $a = P(\text{HH beats TT})$ and $1 - a = P(\text{TT beats HH})$, then backtracking gives us diagram (c) below. This gives us the equations $3a - 1 = \frac{1}{2}$ and $2 - 3a = \frac{1}{2}$, both of which yield the solution $a = \frac{1}{2}$.
- $P(\text{HT beats TH}) = 1/2$: If we let $a = P(\text{HT beats TH})$ and $1 - a = P(\text{TH beats HT})$, then backtracking gives us diagram (d) below. This gives us the equations $a = \frac{1}{2}$ and $1 - a = \frac{1}{2}$, both of which yield the solution $a = \frac{1}{2}$.

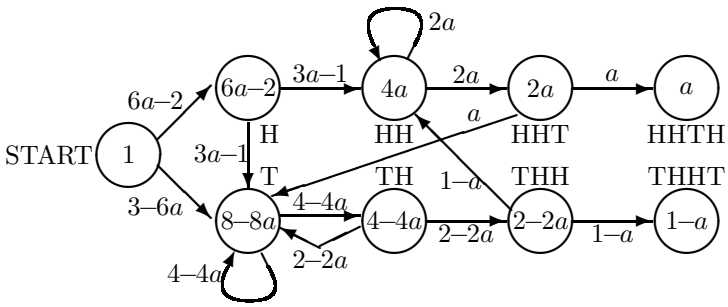


The remaining entries in the table can be completed using complements and symmetries:

- $P(\text{HT beats HH}) = 1 - 1/2 = 1/2$: Complement of HH beats HT.
- $P(\text{TH beats HH}) = 1 - 1/4 = 3/4$: Complement of HH beats TH.
- $P(\text{TT beats HH}) = 1/2 (= 1 - 1/2)$: Symmetric to (and complement of) HH beats TT.

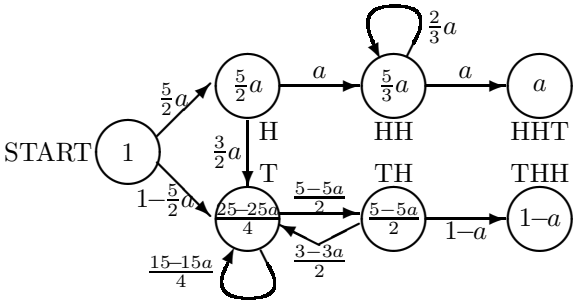
- $P(\text{HT beats TT}) = 3/4$: Symmetric to TH beats HH.
- $P(\text{TH beats HT}) = 1/2 (= 1 - 1/2)$: Symmetric to (and complement of) HT beats TH.
- $P(\text{TH beats TT}) = 1/2$: Symmetric to HT beats HH.
- $P(\text{TT beats HT}) = 1 - 3/4 = 1/4$: Complement of HT beats TT.
- $P(\text{TT beats TH}) = 1 - 1/2 = 1/2$: Complement of TH beats TT.

Exercise 4, Page 15 - If we let $a = P(\text{HHTH beats THHT})$ and $1-a = P(\text{THHT beats HHTH})$, then backtracking gives us the diagram below:



Since heads and tails are equally likely on each toss, we know that the probability of heads and the probability of tails on the first toss will both be equal to 1/2. Solving either of the two equations $6a - 2 = \frac{1}{2}$ or $3 - 6a = \frac{1}{2}$ yields the solution $a = 5/12$.

Exercise 5, Page 16 - When we backtrack in this game we must remember that 2/5 of the probability mass flowing out of each node goes toward heads and 3/5 goes toward tails on the next toss. This gives us the diagram below:



Since the probability of heads is 2/5 and the probability of tails is 3/5, we get the equations $\frac{5}{2}a = \frac{2}{5}$ and $1 - \frac{5}{2}a = \frac{3}{5}$, both of which yield the solution $a = 4/25$. Thus, the probability of sequence HHT beating sequence THH in this game is 4/25.

Exercise 6, Page 18 - Since the probability of losing is 4/5 with a payoff of −\$1 (you pay \$1 to play and get nothing back if you lose for a net payoff of −\$1) and the probability of winning is therefore 1/5 with a payoff of \$4 (you pay \$1 to play and get \$5 back if you win for a net payoff of \$4), the amount of money we expect to have after completing this game is $\frac{4}{5}(-\$1) + \frac{1}{5}(\$4) = \$0$.

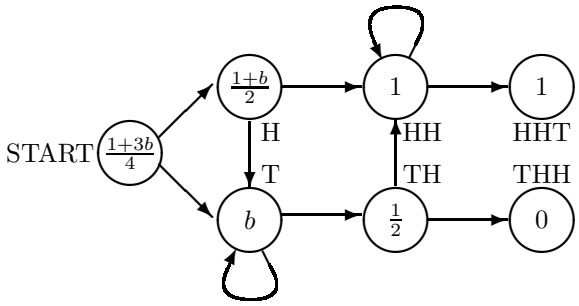
Exercise 7, Page 18 - The expected number of tosses is given by the sum of the values on all arrows in part (b) of the diagram in the solution to Exercise 3. In that exercise we determined that $a = 1/4$ is the probability of HH beating TH. Using that value of a gives us a sum of 3. We expect to use three tosses to complete this game.

Exercise 8, Page 18 - The expected number of tosses is given by the sum of the values on all arrows in the diagram in the solution to Exercise 4. In that exercise we determined that $a = 5/12$ is the probability of HHTH beating THHT. Using that value of a gives us a sum of $73/6$ or $12\frac{1}{6}$ expected tosses.

Exercise 9, Page 18 - The expected number of tosses is given by the sum of the values on all arrows in the diagram in the solution to Exercise 5. In that exercise we determined that $a = 4/25$ is the probability of HHT beating THH when the probability of a head is $2/5$ and the probability of a tail is $3/5$. Using that value of a gives us a sum of $541/60$ or $9\frac{1}{60}$ expected tosses. This differs from the expected number of tosses when the coin is fair (computed in Section 2 to be 6.5 tosses).

Section 3 Solutions

Exercise 1, Page 21 - This game is the same as the one in Exercise 1, Page 15. The diagram is given below.



Nodes HHT and THT: Prob(HHT|HHT) = 1 is obvious, as is Prob(HHT|THT) = 0.

Node HH:

$$\begin{aligned} \text{Prob(HHT|HH)} &= \text{Prob}(nf = \text{H})\text{Prob(HHT|HH)} + \text{Prob}(nf = \text{T})\text{Prob(HHT|HHT)} \\ &= \frac{1}{2}\text{Prob(HHT|HH)} + \frac{1}{2}(1) \end{aligned}$$

Solving for Prob(HHT|HH) gives us Prob(HHT|HH) = 1.

Node TH:

$$\begin{aligned} \text{Prob(HHT|TH)} &= \text{Prob}(nf = \text{H})\text{Prob(HHT|HH)} + \text{Prob}(nf = \text{T})\text{Prob(HHT|THT)} \\ &= \frac{1}{2}(1) + \frac{1}{2}(0) \\ &= \frac{1}{2} \end{aligned}$$

Node H:

$$\begin{aligned} \text{Prob(HHT|H)} &= \text{Prob}(nf = \text{H})\text{Prob(HHT|HH)} + \text{Prob}(nf = \text{T})\text{Prob(HHT|T)} \\ &= \frac{1}{2}(1) + \frac{1}{2}(\text{Prob(HHT|T)}) \end{aligned}$$

If we let $b = \text{Prob(HHT|T)}$, we get that $\text{Prob(HHT|H)} = \frac{1+b}{2}$. We will return to solve this once we have determined the value of b.

START Node:

$$\begin{aligned}\text{Prob(HHT|START)} &= \text{Prob}(nf = \text{H})\text{Prob(HHT|H)} + \text{Prob}(nf = \text{T})\text{Prob(HHT|T)} \\ &= \frac{1}{2}\left(\frac{1+b}{2}\right) + \frac{1}{2}(b)) \\ &= \frac{1+3b}{4}\end{aligned}$$

Now we get that Prob(HHT|START) $= \frac{1+3b}{4}$. We will return to solve this once we have determined the value of b.

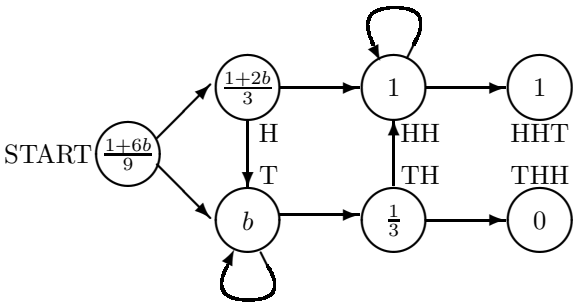
Node T:

$$\begin{aligned}\text{Prob(HHT|T)} &= \text{Prob}(nf = \text{H})\text{Prob(HHT|TH)} + \text{Prob}(nf = \text{T})\text{Prob(HHT|T)} \\ &= \frac{1}{2}\left(\frac{1}{2}\right) + \frac{1}{2}(b))\end{aligned}$$

This gives us $\frac{b}{2} = \frac{1}{4}$, or $b = \frac{1}{2}$.

Thus, Prob(HHT|T) $= \frac{1}{2}$, Prob(HHT|H) $= \frac{1+b}{2} = \frac{3}{4}$,and Prob(HHT|START) $= \frac{1+3b}{4} = \frac{5}{8}$.

Exercise 2, Page 21 - This game has the diagram as the one for Exercise 1, Page 21. The diagram is given below.



Nodes HHT and THT: Prob(HHT|HHT) $= 1$ is obvious, as is Prob(HHT|THT) $= 0$.

Node HH:

$$\begin{aligned}\text{Prob(HHT|HH)} &= \text{Prob}(nf = \text{H})\text{Prob(HHT|HH)} + \text{Prob}(nf = \text{T})\text{Prob(HHT|HHT)} \\ &= \frac{1}{3}\text{Prob(HHT|HH)} + \frac{2}{3}(1)\end{aligned}$$

Solving for Prob(HHT|HH) gives us Prob(HHT|HH) $= 1$.

Node TH:

$$\begin{aligned}\text{Prob(HHT|TH)} &= \text{Prob}(nf = \text{H})\text{Prob(HHT|HH)} + \text{Prob}(nf = \text{T})\text{Prob(HHT|THT)} \\ &= \frac{1}{3}(1) + \frac{2}{3}(0) \\ &= \frac{1}{3}\end{aligned}$$

Node H:

$$\begin{aligned}\text{Prob(HHT|H)} &= \text{Prob}(nf = \text{H})\text{Prob(HHT|HH)} + \text{Prob}(nf = \text{T})\text{Prob(HHT|T)} \\ &= \frac{1}{3}(1) + \frac{2}{3}(\text{Prob(HHT|T)})\end{aligned}$$

If we let $b = \text{Prob(HHT|T)}$, we get that $\text{Prob(HHT|H)} = \frac{1+2b}{3}$. We will return to solve this once we have determined the value of b.

START Node:

$$\begin{aligned}\text{Prob(HHT|START)} &= \text{Prob}(nf = \text{H})\text{Prob(HHT|H)} + \text{Prob}(nf = \text{T})\text{Prob(HHT|T)} \\ &= \frac{1}{3}\left(\frac{1+2b}{3}\right) + \frac{2}{3}(b)) \\ &= \frac{1+6b}{9}\end{aligned}$$

Now we get that $\text{Prob(HHT|START)} = \frac{1+6b}{9}$. We will return to solve this once we have determined the value of b.

Node T:

$$\begin{aligned}\text{Prob(HHT|T)} &= \text{Prob}(nf = \text{H})\text{Prob(HHT|TH)} + \text{Prob}(nf = \text{T})\text{Prob(HHT|T)} \\ &= \frac{1}{3}\left(\frac{1}{3}\right) + \frac{2}{3}(b))\end{aligned}$$

Since $b = \text{Prob(HHT|T)}$, this gives us $b = \frac{1}{9} + \frac{2}{3}b$, or $b = \frac{1}{3}$.

Thus, $\text{Prob(HHT}|T) = \frac{1}{3}$, $\text{Prob(HHT|H)} = \frac{1+2b}{3} = \frac{5}{9}$,and $\text{Prob(HHT|START)} = \frac{1+6b}{9} = \frac{11}{27}$.

Section 4 Solutions

Exercise 1, Page 25 - Refer to the diagram in the solution to Exercise 1, Page 15 (without regarding the backfiring analysis). Starting from the critically loaded configuration, if eight chips are added to the START node we will return to the critically loaded configuration, with five chips on node HHT and three chips on node THT. This implies that the probability of node HHT winning is 5/8, which confirms our analysis in Exercise 1, Page 15.

Exercise 2, Page 25 - Refer to the diagram in the solution to Exercise 2, Page 15 (without regarding the backfiring analysis). Starting from the critically loaded configuration, if five chips are added to the \$1 node we will return to the critically loaded configuration, with four chips on the \$0 node and one chip on the \$5 node. This implies that the probability of winning Lotto Game Two is 1/5, which confirms our analysis in Exercise 2, Page 15.

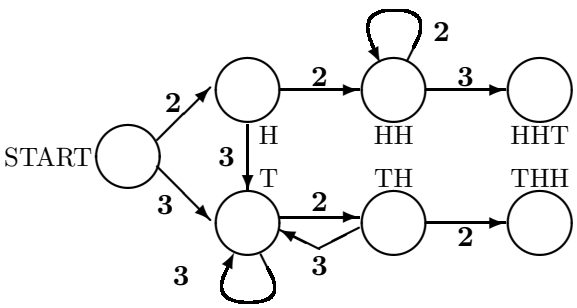
Exercise 3, Page 25 - We need only verify the four entries in the table that were computed using a backtracking analysis. The remaining entries can be completed by the same symmetry/complement analysis that was done in Exercise 3, Page 15.

- Refer to part (a) of the diagram in the solution to Exercise 3, Page 15 (without regarding the backfiring analysis). Starting from the critically loaded configuration, if two chips are added to the START node we will return to the critically loaded configuration, with one chip on node HH and one chip on node HT. This implies that the probability of HH beating HT is 1/2, which confirms our analysis in Exercise 3, Page 15.

- Refer to part (b) of the diagram in the solution to Exercise 3, Page 15 (without regarding the backfiring analysis). Starting from the critically loaded configuration, if four chips are added to the START node we will return to the critically loaded configuration, with one chip on node HH and three chips on node TH. This implies that the probability of HH beating TH is 1/4, which confirms our analysis in Exercise 3, Page 15.
- Refer to part (c) of the diagram in the solution to Exercise 3, Page 15 (without regarding the backfiring analysis). Starting from the critically loaded configuration, if two chips are added to the START node we will return to the critically loaded configuration, with one chip on node HH and one chip on node TT. This implies that the probability of HH beating TT is 1/2, which confirms our analysis in Exercise 3, Page 15.
- Refer to part (d) of the diagram in the solution to Exercise 3, Page 15 (without regarding the backfiring analysis). Starting from the critically loaded configuration, if two chips are added to the START node we will return to the critically loaded configuration, with one chip on node HT and one chip on node TH. This implies that the probability of HT beating TH is 1/2, which confirms our analysis in Exercise 3, Page 15.

Exercise 4, Page 25 - Refer to part (b) of the diagram in the solution to Exercise 3, Page 15 (without regarding the backfiring analysis). Starting from the critically loaded configuration, if four chips are added to the START node we will return to the critically loaded configuration, with one chip on node HH and three chips on node TH. It requires a total of 12 moves to return to the critically loaded configuration. This implies that the expected number of tosses to complete a game between HH and TH is $\frac{12}{4} = 3$, which confirms our analysis in Exercise 7, Page 18.

Exercise 5, Page 25 - In order to model a coin tossing game in which the probability of tossing a head is 2/5, we could recreate the original graph for this game, but have five arrows leaving each node, two going to the node that results from H on the next toss and three going to the node that results from T on the next toss. This is difficult to draw, so we will represent it here by putting a boldface numeral **2** or **3** on each arrow to indicate the number of arrows we wish to use. The resulting diagram is below:

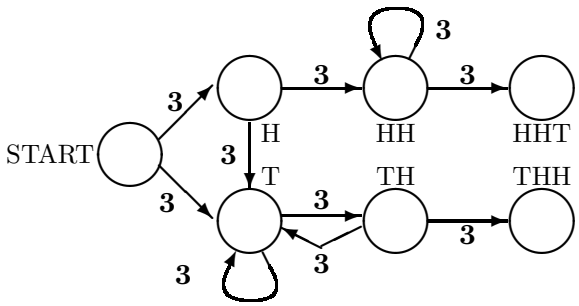


Since each node has five arrows leaving it, the critically loaded configuration will have four chips on each interior node. We will add chips five at a time to the START node, and proceed as usual. The details of this are left to the reader, however we note that the critically loaded configuration recurs after adding 300 chips to the game, and results in 48 chips on node HHT and 252 chips on node THH. Thus, the probability of HHT beating THH in this game is $\frac{48}{300} = \frac{4}{25}$, which confirms our analysis in Exercise 5, Page 16. Further, it took 2705 total moves to return to the critically loaded configuration, so the expected number of moves is $\frac{2705}{300} = \frac{541}{60} = 9\frac{1}{60}$ as we found in Exercise 9, Page 18.

Exercise 6, Page 25 - In each of the four exercises, firing the nodes in a different order should result in exactly the same results as in the original analysis.

Section 5 Solutions

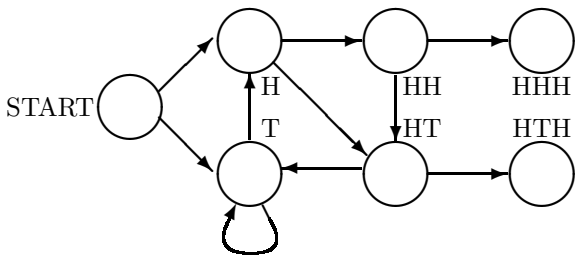
Exercise 1, Page 34 - It is difficult to draw the required diagram, so we will represent it here by putting a boldface numeral **3** on each arrow to indicate that each arrow in the diagram actually represents three arrows. The resulting diagram is below:



Because each node has six arrows leaving it, the critically loaded configuration has five chips on each interior node. This configuration recurs after 12 chips are added to the START node, and it takes a total of 78 moves. When we reach the critically loaded configuration there are three chips on node HHT and nine chips on node THH. Thus, the ratios of chips on nodes HHT and THH to the number of chips added to the START node are 1/4 and 3/4 respectively, and the expected number of tosses to complete the game is $78/12 = 6.5$. These values agree with the ratios we obtained in the original coin tossing game between HHT and THH.

Exercise 2, Page 34 - If we were to multiply the number of arrows between each pair of nodes by a positive integer n, we would expect the critically loaded configuration to recur after n times as many chips had been added to the START node, n times as many moves had been made, and n times as many chips ended up on each absorbing node. Thus, all of the ratios we computed would remain the same.

Exercise 3, Page 34 - The diagram for the coin tossing game between HHH and HTH is given below:



We will let s_1 = START, s_2 = H, s_3 = T, s_4 = HH, s_5 = HT, s_6 = HHH, and s_7 = HTH.

Thus we have the matrices P, Q, and R given by $P = \left(\begin{array}{ccccc|cc} 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right)$,

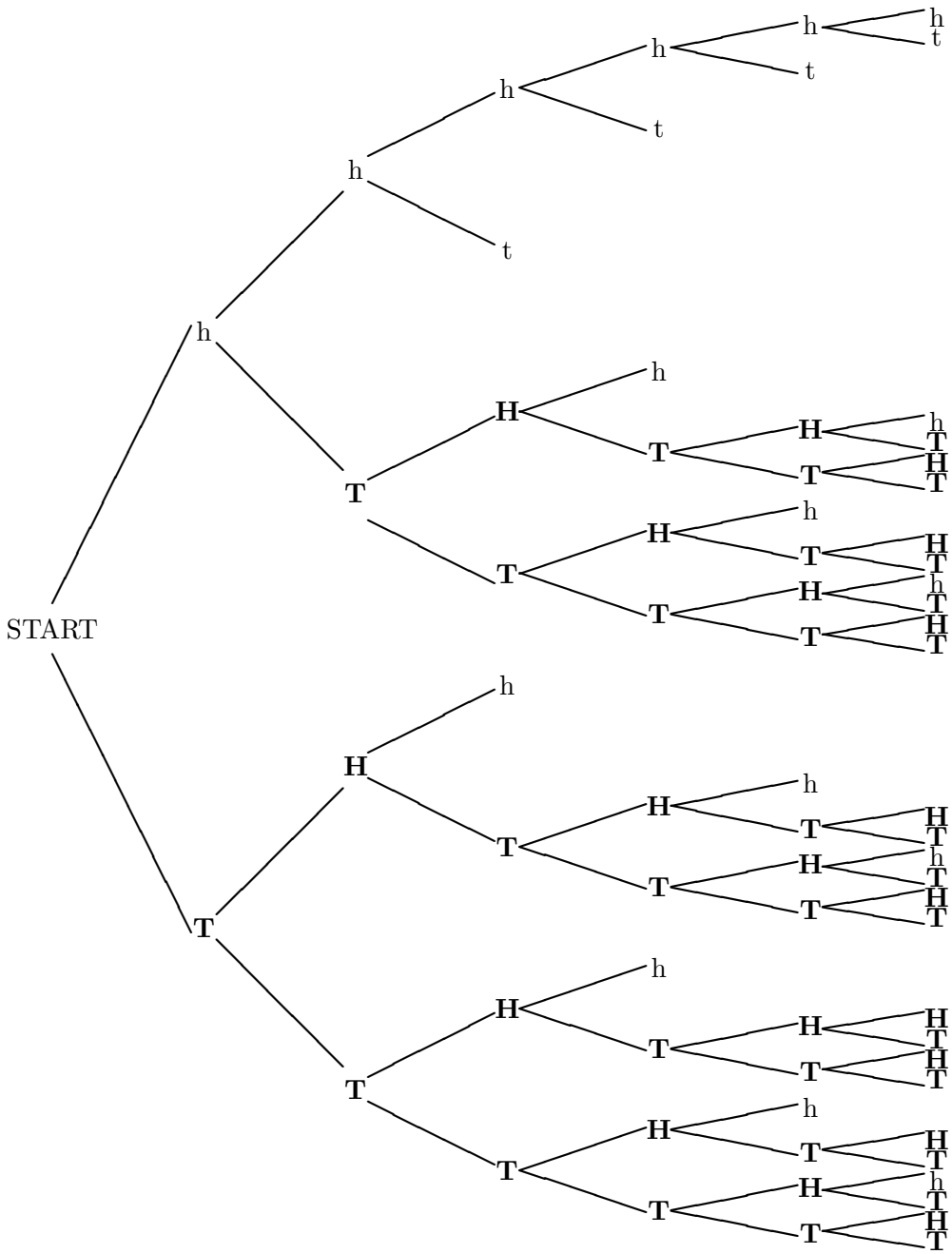
$Q = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 1/2 & 0 & 0 \end{pmatrix}$, and $R = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}$.

The matrix $I - Q$ is therefore given by $I - Q = \begin{pmatrix} 1 & -1/2 & -1/2 & 0 & 0 \\ 0 & 1 & 0 & -1/2 & -1/2 \\ 0 & -1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1/2 \\ 0 & 0 & -1/2 & 0 & 0 \end{pmatrix}$. Using technology we compute $(I - Q)^{-1} = \begin{pmatrix} 1 & 1.6 & 2.2 & 0.8 & 1.2 \\ 0 & 1.6 & 1.2 & 0.8 & 1.2 \\ 0 & 1.6 & 3.2 & 0.8 & 1.2 \\ 0 & 0.4 & 0.8 & 1.2 & 0.8 \\ 0 & 0.8 & 1.6 & 0.4 & 1.6 \end{pmatrix}$.

This gives us that $(I - Q)^{-1}R = \begin{pmatrix} 1 & 1.6 & 2.2 & 0.8 & 1.2 \\ 0 & 1.6 & 1.2 & 0.8 & 1.2 \\ 0 & 1.6 & 3.2 & 0.8 & 1.2 \\ 0 & 0.4 & 0.8 & 1.2 & 0.8 \\ 0 & 0.8 & 1.6 & 0.4 & 1.6 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1/2 & 0 \\ 0 & 1/2 \end{pmatrix} = \begin{pmatrix} 2/5 & 3/5 \\ 2/5 & 3/5 \\ 2/5 & 3/5 \\ 3/5 & 2/5 \\ 1/5 & 4/5 \end{pmatrix}$.

From this we can conclude that the probability of HHH winning the game is 2/5. We can also conclude that the probability of HHH winning doesn't change if we get an H on the first toss. This can also be seen by noting that we must have an H for the first toss of either winning sequence.

Exercise 4, Page 35 - Below is the tree for HHT vs. THH. An uppercase **T** represents state s_3 (node T). An uppercase **H** represents state s_5 (node TH). A lowercase "h" represents either state s_7 (a win for THH), state s_4 (node HH), or state s_2 (node H). A lowercase "t" represents state s_6 (a win for HHT). Note that each occurrence of **H** follows a **T** and that for each **H** following a **T** there is another **T** following the first **T**. Thus, for games starting at state s_1, half of the visits to state s_3 will yield visits to state s_5. That means $t_{15} = \frac{1}{2}t_{13}$.


START
h
T
h
t
H
T
h
t
H
T
h
t
H

References

[1] S. Althoen, L. King, and K. Schilling, *How long is a game of snakes and ladders?*, Mathematical Gazette **77** (1993), no. 478, 71–76.

[2] N. L. Biggs, *Chip-firing and the critical group of a graph*, Journal of Algebraic Combinatorics **9** (1999), 25–45.

[3] Javier Bitar and Eric Goles, *Parallel chip firing game graphs*, Theoretical Computer Science **92** (1992), 291–300.

[4] Anders Björner and László Lovász, *Chip-firing games on directed graphs*, J. Algebraic Combin. **1** (1992), no. 4, 305–328.

[5] Anders Björner, László Lovász, and Peter W. Shor, *Chip-firing games on graphs*, European J. Combin. **12** (1991), no. 4, 283–291.

[6] R. Dirks, *Hi Ho! Cherry-O!, Markov chains, and Mathematica*, Stats (Spring 1999), no. 25, 23–27.

[7] Arthur Engel, *The probabilistic abacus*, Educational Studies in Mathematics **6** (1975), 1–22.

[8] Arthur Engel, *Why does the probabilistic abacus work?*, Educational Studies in Mathematics **7** (1976), 59–69.

[9] S. Gadbois, *Mr. Markov plays Chutes and Ladders*, The UMAP Journal **14** (1993), no. 1, 31–38.

[10] Eric Goles and Marcos A. Kiwi, *Games on line graphs and sand piles*, Theoretical Computer Science **115** (1993), 321–349.

[11] R. Johnson, *The dice game "Drop Dead"*, Teaching Mathematics & Its Applications **15** (1996), no. 3, 97–98.

[12] R. Johnson, *Using games to teach Markov chains*, PRIMUS **13** (2003), no. 4, 337–348.

[13] J. Kemeny and J. Snell, *Finite Markov chains*, D. Van Nostrand, Princeton, NJ, 1960.

[14] Fred S. Roberts, *Discrete mathematical models, with applications to social, biological, and environmental problems*, Prentice-Hall, Englewood Cliffs, NJ, 1976.

[15] Gábor Tardos, *Polynomial bound for a chip firing game on graphs*, SIAM Journal of Discrete Mathematics **1** (1988), no. 3, 397–398.

[16] Jan van den Heuvel, *Algorithmic aspects of a chip-firing game*, Combinatorics, Probability & Computing **10** (2001), no. 6, 505–529.